STM8 Device Headers

Generated by Doxygen 1.8.13

Contents

Chapter 1

Main Page

1.1 Device Overview

This is a set of open-source device headers for the 8-bit STM8 microcontroller family by STMicroelectronics. For a brief introduction to the STM8 see e.g. here or here.

Due to a superb performance to cost ratio, low current consumption and ease-of-use, STM8s are widely used in low-end electronic gadgets, and a lot of development boards are readily available.

1.2 Toolchains

Several commercial cross-compilers support the STM8, namely Cosmic, IAR and Raisonance. In addition, the open-source compiler SDCC also supports the STM8 since v3.4.0.

All commercial compilers come with device headers pre-installed, but these are mutually incompatible and (of course) not open-source. SDCC does not come with STM8 device headers due to license issues.

To facilitate development, STM provides the so-called Standard Peripheral Library (SPL) which contains not only device headers, but also standardized peripherals functions. The SPL is compatible with all above commercial compilers out of the box, and with SDCC via a patch. Unfortunately the SPL is also not 100% open-source, as discussed here or here.

1.3 Open-Source Headers

A lot of (more or less complete) STM8 device headers is available on the internet, e.g. here or here. Alternatively one can also use the SPL headers, or those from the above commercial compilers (if license allows). However, these are all mutually incompatible, and thus hinder the exchange of code snippets between projects. Compare this to the Arduino and avr-libc multiverse... :-(

Following a longer discussion, here is a proposal for open-source STM8 device headers, published under GPL3+. They are intended for inclusion into the SDCC distribution, but are compatible with all above compilers. My final aim is to provide standardized device headers for all available STM8 devices, which can be used in open-source projects without restrictions, and facilitate code exchange withon the STM8 OS-community.

1.3.1 Content

- device headers for all STM8AF, STM8S and STM8L10x devices (76 of 138 STM8 devices and growing)
- this reference as HTML and PDF (created by Doxygen)
- example projects for SDCC, Cosmic, IAR and Raisonance toolchains
  - pin toggle (aka blink)
  - timer interrupts
  - serial communication (incl. printf() and gets())
  - analog measurements
  - PWM generation
  - mixing with SPL headers and routines

1.3.2 Features

- direct/bitwise or byte-wise access to all peripheral registers
- assembler macros for common operations like *NOP()* or *DISABLE_INTERRUPTS()*
- can be mixed with SPL headers and functions (see example *blink_TIM4_SPL*)

Example:

```
// device header (STM8 Discovery board)
#include "STM8S105C6.h"

// configure LED pin to output push-pull (bitmasks)
_GPIOD_DDR |= _GPIO_PIN0;         // input(=0) or output(=1)
_GPIOD_CR1 |= _GPIO_PIN0;         // input: 0=float, 1=pull-up; output: 0=open-drain,
       1=push-pull
_GPIOD_CR2 |= _GPIO_PIN0;         // input: 0=no exint, 1=exint; output: 0=2MHz slope,
       1=10MHz slope

// main loop
while (1) {

  // toggle LED
  //_GPIOD_ODR ^= _GPIO_PIN0;         // byte access (smaller)
  _GPIOD.ODR.PIN0 ^= 1;             // bit access (more convenient)

  // wait a bit
  for (uint16_t i=20000L; i; i--)
    NOP();

} // main loop
```

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 4

File Index

4.1 File List

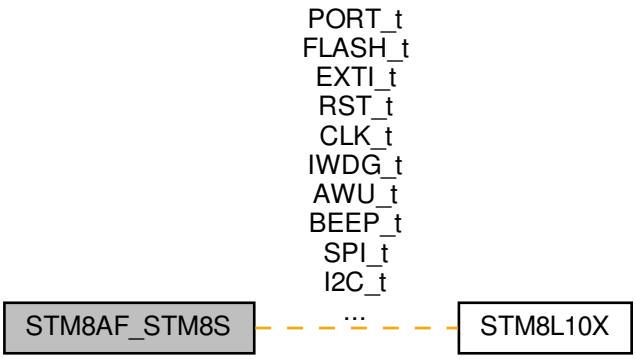
Here is a list of all files with brief descriptions:

Chapter 5

Module Documentation

5.1 STM8AF_STM8S

Collaboration diagram for STM8AF_STM8S:



Data Structures

- struct PORT_t

  structure for controlling pins in PORT mode (PORTx, x=A..I)
- struct FLASH_t

  struct to control write/erase of flash memory (FLASH)
- struct EXTI_t

  struct for configuring external port interrupts (EXTI)
- struct RST_t

  struct for determining reset source (RST)
- struct CLK_t

  struct for configuring/monitoring clock module (CLK)
- struct WWDG_t

struct for access to Window Watchdog registers (WWDG)

- struct IWDG_t

  struct for access to Independent Timeout Watchdog registers (IWDG)
- struct AWU_t

  struct for cofiguring the Auto Wake-Up Module (AWU)
- struct BEEP_t

  struct for beeper control (BEEP)
- struct SPI_t

  struct for controlling SPI module (SPI)
- struct I2C_t

  struct for controlling I2C module (I2C)
- struct UART1_t

  struct for controlling Universal Asynchronous Receiver Transmitter 1 (UART1)
- struct UART2_t

  struct for controlling Universal Asynchronous Receiver Transmitter 2 (UART2)
- struct UART3_t

  struct for controlling Universal Asynchronous Receiver Transmitter 3 (UART3)
- struct UART4_t

  struct for controlling Universal Asynchronous Receiver Transmitter 4 (UART4)
- struct TIM1_t

  struct for controlling 16-Bit Timer 1 (TIM1)
- struct TIM2_t

  struct for controlling 16-Bit Timer 2 (TIM2)
- struct TIM3_t

  struct for controlling 16-Bit Timer 3 (TIM3)
- struct TIM4_t

  struct for controlling 8-Bit Timer 4 (TIM4)
- struct TIM5_t

  struct for controlling 16-Bit Timer 5 (TIM5)
- struct TIM6_t

  struct for controlling 8-Bit Timer 6 (TIM6)
- struct ADC1_t

  struct containing Analog Digital Converter 1 (ADC1)
- struct ADC2_t

  struct containing Analog Digital Converter 2 (ADC2)
- struct CAN_t

  struct for controlling Controller Area Network Module (CAN)
- struct CFG_t

  struct for Global Configuration registers (CFG)
- struct ITC_t

  struct for setting interrupt Priority (ITC)

Macros

- #define STM8AF5268
- #define STM8AF526x
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800

- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF5269
- #define STM8AF526x
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210

- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF5286
- #define STM8AF528x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF5288
- #define STM8AF528x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005

- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF5289
- #define STM8AF528x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240

- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF528A
- #define STM8AF528x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF52A6
- #define STM8AF52Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F

- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF52A8
- #define STM8AF52Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300

- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF52A9
- #define STM8AF52Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF52AA
- #define STM8AF52Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019

- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6213
- #define STM8AF621x
- #define STM8_PFLASH_SIZE 4096
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART4_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6213A

- #define STM8AF621x
- #define STM8_PFLASH_SIZE 4096
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART4_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6223
- #define STM8AF622x
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART4_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340

- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6223A
- #define STM8AF622x
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART4_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6226
- #define STM8AF622x
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200

- #define I2C_AddressBase 0x5210
- #define UART4_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6246
- #define STM8AF624x
- #define STM8_PFLASH_SIZE 16384
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 512
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6248
- #define STM8AF624x
- #define STM8_PFLASH_SIZE 16384
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 512
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A

- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6266
- #define STM8AF626x
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6268
- #define STM8AF626x
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024

- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6269
- #define STM8AF626x
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340

- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6286
- #define STM8AF628x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6288
- #define STM8AF628x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A

- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6289
- #define STM8AF628x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF628A

- #define STM8AF628x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF62A6
- #define STM8AF62Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0

- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF62A8
- #define STM8AF62Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF62A9
- #define STM8AF62Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800

- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF62AA
- #define STM8AF62Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230

- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6366
- #define STM8AF636x
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8AF6388
- #define STM8AF638x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028

- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8_PFLASH_SIZE 2048

  size of program flash [B]
- #define STM8_RAM_SIZE 1024

  size of RAM [B]
- #define STM8_EEPROM_SIZE 128

  size of data EEPROM [B]
- #define STM8_PFLASH_START 0x8000

  first address in program flash
- #define STM8_PFLASH_END (STM8_PFLASH_START + STM8_PFLASH_SIZE - 1)

  last address in program flash
- #define STM8_RAM_START 0x0000

  first address in RAM
- #define STM8_RAM_END (STM8_RAM_START + STM8_RAM_SIZE - 1)

  last address in RAM
- #define STM8_EEPROM_START 0x4000

  first address in EEPROM
- #define STM8_EEPROM_END (STM8_EEPROM_START + STM8_EEPROM_SIZE - 1)

  last address in EEPROM
- #define STM8_ADDR_WIDTH 16

  width of address space
- #define STM8_MEM_POINTER_T uint16_t

  address variable type
- #define ISR_HANDLER(func, irq) void func(void) __interrupt(irq)

  handler for interrupt service routine
- #define ISR_HANDLER_TRAP(func) void func() __trap

  handler for trap service routine
- #define NOP() __asm__("nop")

  perform a nop() operation (=minimum delay)
- #define DISABLE_INTERRUPTS() __asm__("sim")

  disable interrupt handling
- #define ENABLE_INTERRUPTS() __asm__("rim")

  enable interrupt handling

- #define TRIGGER_TRAP __asm__("trap")

  trigger a trap (=soft interrupt) e.g. for EMC robustness (see AN1015)
- #define WAIT_FOR_INTERRUPT() __asm__("wfi")

  stop code execution and wait for interrupt
- #define ENTER_HALT() __asm__("halt")

  put controller to HALT mode
- #define SW_RESET() (_WWDG_CR=0xBF)

  reset controller via WWGD module
- #define _BITS unsigned int

  data type in bit structs (follow C90 standard)
- #define _SFR(type, addr) (*((volatile type*) (addr)))

  peripheral register
- #define __TLI_VECTOR__ 0

  irq0 - External Top Level interrupt (TLI) for pin PD7
- #define __AWU_VECTOR__ 1

  irq1 - Auto Wake Up from Halt interrupt (AWU)
- #define __CLK_VECTOR__ 2

  irq2 - Clock Controller interrupt
- #define __PORTA_VECTOR__ 3

  irq3 - External interrupt 0 (GPIOA)
- #define __PORTB_VECTOR__ 4

  irq4 - External interrupt 1 (GPIOB)
- #define __PORTC_VECTOR__ 5

  irq5 - External interrupt 2 (GPIOC)
- #define __PORTD_VECTOR__ 6

  irq6 - External interrupt 3 (GPIOD)
- #define __PORTE_VECTOR__ 7

  irq7 - External interrupt 4 (GPIOE)
- #define __CAN_RX_VECTOR__ 8

  irq8 - CAN receive interrupt (shared with __PORTF_VECTOR__)
- #define __PORTF_VECTOR__ 8

  irq8 - External interrupt 5 (GPIOF, shared with __CAN_RX_VECTOR__)
- #define __CAN_TX_VECTOR__ 9

  irq9 - CAN transmit interrupt
- #define __SPI_VECTOR__ 10

  irq10 - SPI End of transfer interrupt
- #define __TIM1_UPD_OVF_VECTOR__ 11

  irq11 - TIM1 Update/Overflow/Trigger/Break interrupt
- #define __TIM1_CAPCOM_VECTOR__ 12

  irq12 - TIM1 Capture/Compare interrupt
- #define __TIM2_UPD_OVF_VECTOR__ 13

  irq13 - TIM2 Update/overflow interrupt (shared with __TIM5_UPD_OVF_VECTOR__)
- #define __TIM5_UPD_OVF_VECTOR__ 13

  irq13 - TIM5 Update/overflow interrupt (shared with __TIM2_UPD_OVF_VECTOR__)
- #define __TIM2_CAPCOM_VECTOR__ 14

  irq14 - TIM2 Capture/Compare interrupt (shared with __TIM5_CAPCOM_VECTOR__)
- #define __TIM3_UPD_OVF_VECTOR__ 15

  irq15 - TIM3 Update/overflow interrupt
- #define __TIM3_CAPCOM_VECTOR__ 16

  irq16 - TIM3 Capture/Compare interrupt
- #define __UART1_TXE_VECTOR__ 17

irq17 - USART/UART1 send (TX empty) interrupt

- #define __UART1_RXF_VECTOR__ 18

  irq18 - USART/UART1 receive (RX full) interrupt
- #define __I2C_VECTOR__ 19

  irq19 - I2C interrupt
- #define __UART2_TXE_VECTOR__ 20

  irq20 - UART2 send (TX empty) interrupt (shared with __UART3_TXE_VECTOR__ and __UART4_TXE_VECTOR__)
- #define __UART2_RXF_VECTOR__ 21

  irq21 - UART2 receive (RX full) interrupt (shared with __UART3_RXF_VECTOR__ and __UART4_RXF_VECTOR__)
- #define __ADC1_VECTOR__ 22

  irq22 - ADC1 end of conversion (shared with __ADC2_VECTOR__)
- #define __TIM4_UPD_OVF_VECTOR__ 23

  irq23 - TIM4 Update/Overflow interrupt (shared with __TIM6_UPD_OVF_VECTOR__)
- #define __FLASH_VECTOR__ 24

  irq24 - flash interrupt
- #define _GPIOA _SFR(PORT_t, PORTA_AddressBase)

  port A struct/bit access
- #define _GPIOA_ODR _SFR(uint8_t, PORTA_AddressBase+0x00)

  port A output register
- #define _GPIOA_IDR _SFR(uint8_t, PORTA_AddressBase+0x01)

  port A input register
- #define _GPIOA_DDR _SFR(uint8_t, PORTA_AddressBase+0x02)

  port A direction register
- #define _GPIOA_CR1 _SFR(uint8_t, PORTA_AddressBase+0x03)

  port A control register 1
- #define _GPIOA_CR2 _SFR(uint8_t, PORTA_AddressBase+0x04)

  port A control register 2
- #define _GPIOB _SFR(PORT_t, PORTB_AddressBase)

  port B struct/bit access
- #define _GPIOB_ODR _SFR(uint8_t, PORTB_AddressBase+0x00)

  port B output register
- #define _GPIOB_IDR _SFR(uint8_t, PORTB_AddressBase+0x01)

  port B input register
- #define _GPIOB_DDR _SFR(uint8_t, PORTB_AddressBase+0x02)

  port B direction register
- #define _GPIOB_CR1 _SFR(uint8_t, PORTB_AddressBase+0x03)

  port B control register 1
- #define _GPIOB_CR2 _SFR(uint8_t, PORTB_AddressBase+0x04)

  port B control register 2
- #define _GPIOC _SFR(PORT_t, PORTC_AddressBase)

  port C struct/bit access
- #define _GPIOC_ODR _SFR(uint8_t, PORTC_AddressBase+0x00)

  port C output register
- #define _GPIOC_IDR _SFR(uint8_t, PORTC_AddressBase+0x01)

  port C input register
- #define _GPIOC_DDR _SFR(uint8_t, PORTC_AddressBase+0x02)

  port C direction register
- #define _GPIOC_CR1 _SFR(uint8_t, PORTC_AddressBase+0x03)

  port C control register 1

- #define _GPIOC_CR2 _SFR(uint8_t, PORTC_AddressBase+0x04)
  port C control register 2
- #define _GPIOD _SFR(PORT_t, PORTD_AddressBase)
  port D struct/bit access
- #define _GPIOD_ODR _SFR(uint8_t, PORTD_AddressBase+0x00)
  port D output register
- #define _GPIOD_IDR _SFR(uint8_t, PORTD_AddressBase+0x01)
  port D input register
- #define _GPIOD_DDR _SFR(uint8_t, PORTD_AddressBase+0x02)
  port D direction register
- #define _GPIOD_CR1 _SFR(uint8_t, PORTD_AddressBase+0x03)
  port D control register 1
- #define _GPIOD_CR2 _SFR(uint8_t, PORTD_AddressBase+0x04)
  port D control register 2
- #define _GPIOE _SFR(PORT_t, PORTE_AddressBase)
  port E struct/bit access
- #define _GPIOE_ODR _SFR(uint8_t, PORTE_AddressBase+0x00)
  port E output register
- #define _GPIOE_IDR _SFR(uint8_t, PORTE_AddressBase+0x01)
  port E input register
- #define _GPIOE_DDR _SFR(uint8_t, PORTE_AddressBase+0x02)
  port E direction register
- #define _GPIOE_CR1 _SFR(uint8_t, PORTE_AddressBase+0x03)
  port E control register 1
- #define _GPIOE_CR2 _SFR(uint8_t, PORTE_AddressBase+0x04)
  port E control register 2
- #define _GPIOF _SFR(PORT_t, PORTF_AddressBase)
  port F struct/bit access
- #define _GPIOF_ODR _SFR(uint8_t, PORTF_AddressBase+0x00)
  port F output register
- #define _GPIOF_IDR _SFR(uint8_t, PORTF_AddressBase+0x01)
  port F input register
- #define _GPIOF_DDR _SFR(uint8_t, PORTF_AddressBase+0x02)
  port F direction register
- #define _GPIOF_CR1 _SFR(uint8_t, PORTF_AddressBase+0x03)
  port F control register 1
- #define _GPIOF_CR2 _SFR(uint8_t, PORTF_AddressBase+0x04)
  port F control register 2
- #define _GPIOG _SFR(PORT_t, PORTG_AddressBase)
  port G struct/bit access
- #define _GPIOG_ODR _SFR(uint8_t, PORTG_AddressBase+0x00)
  port G output register
- #define _GPIOG_IDR _SFR(uint8_t, PORTG_AddressBase+0x01)
  port G input register
- #define _GPIOG_DDR _SFR(uint8_t, PORTG_AddressBase+0x02)
  port G direction register
- #define _GPIOG_CR1 _SFR(uint8_t, PORTG_AddressBase+0x03)
  port G control register 1
- #define _GPIOG_CR2 _SFR(uint8_t, PORTG_AddressBase+0x04)
  port G control register 2
- #define _GPIOH _SFR(PORT_t, PORTH_AddressBase)

port H struct/bit access

- #define _GPIOH_ODR _SFR(uint8_t, PORTH_AddressBase+0x00)

  port H output register
- #define _GPIOH_IDR _SFR(uint8_t, PORTH_AddressBase+0x01)

  port H input register
- #define _GPIOH_DDR _SFR(uint8_t, PORTH_AddressBase+0x02)

  port H direction register
- #define _GPIOH_CR1 _SFR(uint8_t, PORTH_AddressBase+0x03)

  port H control register 1
- #define _GPIOH_CR2 _SFR(uint8_t, PORTH_AddressBase+0x04)

  port H control register 2
- #define _GPIOI _SFR(PORT_t, PORTI_AddressBase)

  port I struct/bit access
- #define _GPIOI_ODR _SFR(uint8_t, PORTI_AddressBase+0x00)

  port I output register
- #define _GPIOI_IDR _SFR(uint8_t, PORTI_AddressBase+0x01)

  port I input register
- #define _GPIOI_DDR _SFR(uint8_t, PORTI_AddressBase+0x02)

  port I direction register
- #define _GPIOI_CR1 _SFR(uint8_t, PORTI_AddressBase+0x03)

  port I control register 1
- #define _GPIOI_CR2 _SFR(uint8_t, PORTI_AddressBase+0x04)

  port I control register 2
- #define _GPIO_ODR_RESET_VALUE ((uint8_t) 0x00)

  port output register reset value
- #define _GPIO_DDR_RESET_VALUE ((uint8_t) 0x00)

  port direction register reset value
- #define _GPIO_CR1_RESET_VALUE ((uint8_t) 0x00)

  port control register 1 reset value
- #define _GPIO_CR2_RESET_VALUE ((uint8_t) 0x00)

  port control register 2 reset value
- #define _GPIO_PIN0 ((uint8_t) (0x01 << 0))

  port bit mask for pin 0 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN1 ((uint8_t) (0x01 << 1))

  port bit mask for pin 1 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN2 ((uint8_t) (0x01 << 2))

  port bit mask for pin 2 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN3 ((uint8_t) (0x01 << 3))

  port bit mask for pin 3 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN4 ((uint8_t) (0x01 << 4))

  port bit mask for pin 4 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN5 ((uint8_t) (0x01 << 5))

  port bit mask for pin 5 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN6 ((uint8_t) (0x01 << 6))

  port bit mask for pin 6 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN7 ((uint8_t) (0x01 << 7))

  port bit mask for pin 7 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _FLASH _SFR(FLASH_t, FLASH_AddressBase)

  Flash struct/bit access.
- #define _FLASH_CR1 _SFR(uint8_t, FLASH_AddressBase+0x00)

  Flash control register 1 (FLASH_CR1)

- #define _FLASH_CR2 _SFR(uint8_t, FLASH_AddressBase+0x01)

  Flash control register 2 (FLASH_CR2)
- #define _FLASH_NCR2 _SFR(uint8_t, FLASH_AddressBase+0x02)

  complementary Flash control register 2 (FLASH_NCR2)
- #define _FLASH_FPR _SFR(uint8_t, FLASH_AddressBase+0x03)

  Flash protection register (FLASH_FPR)
- #define _FLASH_NFPR _SFR(uint8_t, FLASH_AddressBase+0x04)

  complementary Flash protection register (FLASH_NFPR)
- #define _FLASH_IAPSR _SFR(uint8_t, FLASH_AddressBase+0x05)

  Flash status register (FLASH_IAPSR)
- #define _FLASH_PUKR _SFR(uint8_t, FLASH_AddressBase+0x08)

  Flash program memory unprotecting key register (FLASH_PUKR)
- #define _FLASH_DUKR _SFR(uint8_t, FLASH_AddressBase+0x0A)

  Data EEPROM unprotection key register (FLASH_DUKR)
- #define _FLASH_CR1_RESET_VALUE ((uint8_t) 0x00)

  Flash control register 1 reset value.
- #define _FLASH_CR2_RESET_VALUE ((uint8_t) 0x00)

  Flash control register 2 reset value.
- #define _FLASH_NCR2_RESET_VALUE ((uint8_t) 0xFF)

  complementary Flash control register 2 reset value
- #define _FLASH_IAPSR_RESET_VALUE ((uint8_t) 0x40)

  Flash status register reset value.
- #define _FLASH_PUKR_RESET_VALUE ((uint8_t) 0x00)

  Flash program memory unprotecting key reset value.
- #define _FLASH_DUKR_RESET_VALUE ((uint8_t) 0x00)

  Data EEPROM unprotection key reset value.
- #define _FLASH_FIX ((uint8_t) (0x01 << 0))

  Fixed Byte programming time [0] (in _FLASH_CR1)
- #define _FLASH_IE ((uint8_t) (0x01 << 1))

  Flash Interrupt enable [0] (in _FLASH_CR1)
- #define _FLASH_AHALT ((uint8_t) (0x01 << 2))

  Power-down in Active-halt mode [0] (in _FLASH_CR1)
- #define _FLASH_HALT ((uint8_t) (0x01 << 3))

  Power-down in Halt mode [0] (in _FLASH_CR1)
- #define _FLASH_PRG ((uint8_t) (0x01 << 0))

  Standard block programming [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_FPRG ((uint8_t) (0x01 << 4))

  Fast block programming [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_ERASE ((uint8_t) (0x01 << 5))

  Block erasing [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_WPRG ((uint8_t) (0x01 << 6))

  Word programming [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_OPT ((uint8_t) (0x01 << 7))

  Write option bytes [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_WPB ((uint8_t) (0x3F << 0))

  User boot code area protection bits [5:0] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WPB0 ((uint8_t) (0x01 << 0))

  User boot code area protection bit [0] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WPB1 ((uint8_t) (0x01 << 1))

  User boot code area protection bit [1] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WPB2 ((uint8_t) (0x01 << 2))

User boot code area protection bit [2] (in _FLASH_FPR and _FLASH_NFPR)

- #define _FLASH_WPB3 ((uint8_t) (0x01 << 3))

  User boot code area protection bit [3] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WPB4 ((uint8_t) (0x01 << 4))

  User boot code area protection bit [4] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WPB5 ((uint8_t) (0x01 << 5))

  User boot code area protection bit [5] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WR_PG_DIS ((uint8_t) (0x01 << 0))

  Write attempted to protected page flag [0] (in _FLASH_IAPSR)
- #define _FLASH_PUL ((uint8_t) (0x01 << 1))

  Flash Program memory unlocked flag [0] (in _FLASH_IAPSR)
- #define _FLASH_EOP ((uint8_t) (0x01 << 2))

  End of programming (write or erase operation) flag [0] (in _FLASH_IAPSR)
- #define _FLASH_DUL ((uint8_t) (0x01 << 3))

  Data EEPROM area unlocked flag [0] (in _FLASH_IAPSR)
- #define _FLASH_HVOFF ((uint8_t) (0x01 << 5))

  End of high voltage flag [0] (in _FLASH_IAPSR)
- #define _EXTI _SFR(EXTI_t, EXTI_AddressBase)

  External interrupt struct/bit access.
- #define _EXTI_CR1 _SFR(uint8_t, EXTI_AddressBase+0x00)

  External interrupt control register 1 (EXTI_CR1)
- #define _EXTI_CR2 _SFR(uint8_t, EXTI_AddressBase+0x01)

  External interrupt control register 2 (EXTI_CR2)
- #define _EXTI_CR1_RESET_VALUE ((uint8_t) 0x00)

  External interrupt control register 1 reset value.
- #define _EXTI_CR2_RESET_VALUE ((uint8_t) 0x00)

  External interrupt control register 2 reset value.
- #define _EXTI_PAIS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Port A [1:0] (in _EXTI_CR1)
- #define _EXTI_PAIS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Port A [0] (in _EXTI_CR1)
- #define _EXTI_PAIS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Port A [1] (in _EXTI_CR1)
- #define _EXTI_PBIS ((uint8_t) (0x03 << 2))

  External interrupt sensitivity for Port B [1:0] (in _EXTI_CR1)
- #define _EXTI_PBIS0 ((uint8_t) (0x01 << 2))

  External interrupt sensitivity for Port B [0] (in _EXTI_CR1)
- #define _EXTI_PBIS1 ((uint8_t) (0x01 << 3))

  External interrupt sensitivity for Port B [1] (in _EXTI_CR1)
- #define _EXTI_PCIS ((uint8_t) (0x03 << 4))

  External interrupt sensitivity for Port C [1:0] (in _EXTI_CR1)
- #define _EXTI_PCIS0 ((uint8_t) (0x01 << 4))

  External interrupt sensitivity for Port C [0] (in _EXTI_CR1)
- #define _EXTI_PCIS1 ((uint8_t) (0x01 << 5))

  External interrupt sensitivity for Port C [1] (in _EXTI_CR1)
- #define _EXTI_PDIS ((uint8_t) (0x03 << 6))

  External interrupt sensitivity for Port D [1:0] (in _EXTI_CR1)
- #define _EXTI_PDIS0 ((uint8_t) (0x01 << 6))

  External interrupt sensitivity for Port D [0] (in _EXTI_CR1)
- #define _EXTI_PDIS1 ((uint8_t) (0x01 << 7))

  External interrupt sensitivity for Port D [1] (in _EXTI_CR1)

- #define _EXTI_PEIS ((uint8_t) (0x03 << 0))

  Port E external interrupt sensitivity bits [1:0] (in _EXTI_CR2)
- #define _EXTI_PEIS0 ((uint8_t) (0x01 << 0))

  Port E external interrupt sensitivity bits [0] (in _EXTI_CR2)
- #define _EXTI_PEIS1 ((uint8_t) (0x01 << 1))

  Port E external interrupt sensitivity bits [1] (in _EXTI_CR2)
- #define _EXTI_TLIS ((uint8_t) (0x01 << 2))

  Top level interrupt sensitivity [0] (in _EXTI_CR2)
- #define _RST _SFR(RST_t, RST_AddressBase)

  Reset module struct/bit access.
- #define _RST_SR _SFR(uint8_t, RST_AddressBase+0x00)

  Reset module status register (RST_SR)
- #define _RST_WWDGF ((uint8_t) (0x01 << 0))

  Window Watchdog reset flag [0] (in _RST_SR)
- #define _RST_IWDGF ((uint8_t) (0x01 << 1))

  Independent Watchdog reset flag [0] (in _RST_SR)
- #define _RST_ILLOPF ((uint8_t) (0x01 << 2))

  Illegal opcode reset flag [0] (in _RST_SR)
- #define _RST_SWIMF ((uint8_t) (0x01 << 3))

  SWIM reset flag [0] (in _RST_SR)
- #define _RST_EMCF ((uint8_t) (0x01 << 4))

  EMC reset flag [0] (in _RST_SR)
- #define _CLK _SFR(CLK_t, CLK_AddressBase)

  Clock module struct/bit access.
- #define _CLK_ICKR _SFR(uint8_t, CLK_AddressBase+0x00)

  Internal clock register.
- #define _CLK_ECKR _SFR(uint8_t, CLK_AddressBase+0x01)

  External clock register.
- #define _CLK_CMSR _SFR(uint8_t, CLK_AddressBase+0x03)

  Clock master status register.
- #define _CLK_SWR _SFR(uint8_t, CLK_AddressBase+0x04)

  Clock master switch register.
- #define _CLK_SWCR _SFR(uint8_t, CLK_AddressBase+0x05)

  Clock switch control register.
- #define _CLK_CKDIVR _SFR(uint8_t, CLK_AddressBase+0x06)

  Clock divider register.
- #define _CLK_PCKENR1 _SFR(uint8_t, CLK_AddressBase+0x07)

  Peripheral clock gating register 1.
- #define _CLK_CSSR _SFR(uint8_t, CLK_AddressBase+0x08)

  Clock security system register.
- #define _CLK_CCOR _SFR(uint8_t, CLK_AddressBase+0x09)

  Configurable clock output register.
- #define _CLK_PCKENR2 _SFR(uint8_t, CLK_AddressBase+0x0A)

  Peripheral clock gating register 2.
- #define _CLK_HSITRIMR _SFR(uint8_t, CLK_AddressBase+0x0C)

  HSI clock calibration trimming register.
- #define _CLK_SWIMCCR _SFR(uint8_t, CLK_AddressBase+0x0D)

  SWIM clock control register.
- #define _CLK_ICKR_RESET_VALUE ((uint8_t) 0x01)

  Internal clock register reset value.
- #define _CLK_ECKR_RESET_VALUE ((uint8_t) 0x00)

External clock register reset value.

- #define _CLK_CMSR_RESET_VALUE ((uint8_t) 0xE1)

  Clock master status reset value.
- #define _CLK_SWR_RESET_VALUE ((uint8_t) 0xE1)

  Clock master switch reset value.
- #define _CLK_SWCR_RESET_VALUE ((uint8_t) 0x00)

  Clock switch control reset value.
- #define _CLK_CKDIVR_RESET_VALUE ((uint8_t) 0x18)

  Clock divider register reset value.
- #define _CLK_PCKENR1_RESET_VALUE ((uint8_t) 0xFF)

  Peripheral clock gating register 1 reset value.
- #define _CLK_PCKENR2_RESET_VALUE ((uint8_t) 0xFF)

  Peripheral clock gating register 2 reset value.
- #define _CLK_CSSR_RESET_VALUE ((uint8_t) 0x00)

  Clock security system register reset value.
- #define _CLK_CCOR_RESET_VALUE ((uint8_t) 0x00)

  Configurable clock output register reset value.
- #define _CLK_HSITRIMR_RESET_VALUE ((uint8_t) 0x00)

  HSI clock calibration trimming register reset value.
- #define _CLK_SWIMCCR_RESET_VALUE ((uint8_t) 0x00)

  SWIM clock control register reset value.
- #define _CLK_HSIEN ((uint8_t) (0x01 << 0))

  High speed internal RC oscillator enable [0] (in _CLK_ICKR)
- #define _CLK_HSIRDY ((uint8_t) (0x01 << 1))

  High speed internal oscillator ready [0] (in _CLK_ICKR)
- #define _CLK_FHWU ((uint8_t) (0x01 << 2))

  Fast wakeup from Halt/Active-halt modes [0] (in _CLK_ICKR)
- #define _CLK_LSIEN ((uint8_t) (0x01 << 3))

  Low speed internal RC oscillator enable [0] (in _CLK_ICKR)
- #define _CLK_LSIRDY ((uint8_t) (0x01 << 4))

  Low speed internal oscillator ready [0] (in _CLK_ICKR)
- #define _CLK_REGAH ((uint8_t) (0x01 << 5))

  Regulator power off in Active-halt mode [0] (in _CLK_ICKR)
- #define _CLK_HSEEN ((uint8_t) (0x01 << 0))

  High speed external crystal oscillator enable [0] (in _CLK_ECKR)
- #define _CLK_ECKR_HSERDY ((uint8_t) (0x01 << 1))

  High speed external crystal oscillator ready [0] (in _CLK_ECKR)
- #define _CLK_SWI_HSI ((uint8_t) 0xE1)

  write to CLK_SWR for HSI clock (in _CLK_SWR)
- #define _CLK_SWI_LSI ((uint8_t) 0xD2)

  write to CLK_SWR for LSI clock (in _CLK_SWR)
- #define _CLK_SWI_HSE ((uint8_t) 0xB4)

  write to CLK_SWR for HSE clock (in _CLK_SWR)
- #define _CLK_SWBSY ((uint8_t) (0x01 << 0))

  Switch busy flag [0] (in _CLK_SWCR)
- #define _CLK_SWEN ((uint8_t) (0x01 << 1))

  Switch start/stop enable [0] (in _CLK_SWCR)
- #define _CLK_SWIEN ((uint8_t) (0x01 << 2))

  Clock switch interrupt enable [0] (in _CLK_SWCR)
- #define _CLK_SWIF ((uint8_t) (0x01 << 3))

  Clock switch interrupt flag [0] (in _CLK_SWCR)

- #define _CLK_CPUDIV ((uint8_t) (0x07 << 0))
  CPU clock prescaler [2:0] (in _CLK_CKDIVR)
- #define _CLK_CPUDIV0 ((uint8_t) (0x01 << 0))
  CPU clock prescaler [0] (in _CLK_CKDIVR)
- #define _CLK_CPUDIV1 ((uint8_t) (0x01 << 1))
  CPU clock prescaler [1] (in _CLK_CKDIVR)
- #define _CLK_CPUDIV2 ((uint8_t) (0x01 << 2))
  CPU clock prescaler [2] (in _CLK_CKDIVR)
- #define _CLK_HSIDIV ((uint8_t) (0x03 << 3))
  High speed internal clock prescaler [1:0] (in _CLK_CKDIVR)
- #define _CLK_HSIDIV0 ((uint8_t) (0x01 << 3))
  High speed internal clock prescaler [0] (in _CLK_CKDIVR)
- #define _CLK_HSIDIV1 ((uint8_t) (0x01 << 4))
  High speed internal clock prescaler [1] (in _CLK_CKDIVR)
- #define _CLK_I2C ((uint8_t) (0x01 << 0))
  clock enable I2C [0] (in _CLK_PCKENR1)
- #define _CLK_SPI ((uint8_t) (0x01 << 1))
  clock enable SPI [0] (in _CLK_PCKENR1)
- #define _CLK_UART1 ((uint8_t) (0x01 << 2))
  clock enable UART1 [0] (in _CLK_PCKENR1)
- #define _CLK_UART2 ((uint8_t) (0x01 << 3))
  clock enable UART2 [0] (in _CLK_PCKENR1)
- #define _CLK_TIM4_TIM6 ((uint8_t) (0x01 << 4))
  clock enable TIM4/TIM6 [0] (in _CLK_PCKENR1)
- #define _CLK_TIM2_TIM5 ((uint8_t) (0x01 << 5))
  clock enable TIM2/TIM5 [0] (in _CLK_PCKENR1)
- #define _CLK_TIM3 ((uint8_t) (0x01 << 6))
  clock enable TIM3 [0] (in _CLK_PCKENR1)
- #define _CLK_TIM1 ((uint8_t) (0x01 << 7))
  clock enable TIM1 [0] (in _CLK_PCKENR1)
- #define _CLK_CSSEN ((uint8_t) (0x01 << 0))
  Clock security system enable [0] (in _CLK_CSSR)
- #define _CLK_AUX ((uint8_t) (0x01 << 1))
  Auxiliary oscillator connected to master clock [0] (in _CLK_CSSR)
- #define _CLK_CSSDIE ((uint8_t) (0x01 << 2))
  Clock security system detection interrupt enable [0] (in _CLK_CSSR)
- #define _CLK_CSSD ((uint8_t) (0x01 << 3))
  Clock security system detection [0] (in _CLK_CSSR)
- #define _CLK_CCOEN ((uint8_t) (0x01 << 0))
  Configurable clock output enable [0] (in _CLK_CCOR)
- #define _CLK_CCOSEL ((uint8_t) (0x0F << 1))
  Configurable clock output selection [3:0] (in _CLK_CCOR)
- #define _CLK_CCOSEL0 ((uint8_t) (0x01 << 1))
  Configurable clock output selection [0] (in _CLK_CCOR)
- #define _CLK_CCOSEL1 ((uint8_t) (0x01 << 2))
  Configurable clock output selection [1] (in _CLK_CCOR)
- #define _CLK_CCOSEL2 ((uint8_t) (0x01 << 3))
  Configurable clock output selection [2] (in _CLK_CCOR)
- #define _CLK_CCOSEL3 ((uint8_t) (0x01 << 4))
  Configurable clock output selection [3] (in _CLK_CCOR)
- #define _CLK_CCORDY ((uint8_t) (0x01 << 5))

Configurable clock output ready [0] (in _CLK_CCOR)

- #define _CLK_CCOBSY ((uint8_t) (0x01 << 6))

  Configurable clock output busy [0] (in _CLK_CCOR)
- #define _CLK_AWU ((uint8_t) (0x01 << 2))

  clock enable AWU [0] (in _CLK_PCKENR2)
- #define _CLK_ADC ((uint8_t) (0x01 << 3))

  clock enable ADC [0] (in _CLK_PCKENR2)
- #define _CLK_CAN ((uint8_t) (0x01 << 7))

  clock enable CAN [0] (in _CLK_PCKENR2)
- #define _CLK_HSITRIM ((uint8_t) (0x0F << 0))

  HSI trimming value (some devices only support 3 bits, see DS!) [3:0] (in _CLK_HSITRIMR)
- #define _CLK_HSITRIM0 ((uint8_t) (0x01 << 0))

  HSI trimming value [0] (in _CLK_HSITRIMR)
- #define _CLK_HSITRIM1 ((uint8_t) (0x01 << 1))

  HSI trimming value [1] (in _CLK_HSITRIMR)
- #define _CLK_HSITRIM2 ((uint8_t) (0x01 << 2))

  HSI trimming value [2] (in _CLK_HSITRIMR)
- #define _CLK_HSITRIM3 ((uint8_t) (0x01 << 3))

  HSI trimming value [3] (in _CLK_HSITRIMR)
- #define _CLK_SWIMCLK ((uint8_t) (0x01 << 0))

  SWIM clock divider [0] (in _CLK_SWIMCCR)
- #define _WWDG _SFR(WWDG_t, WWDG_AddressBase)

  Window Watchdog struct/bit access.
- #define _WWDG_CR _SFR(uint8_t, WWDG_AddressBase+0x00)

  Window Watchdog Control register (WWDG_CR)
- #define _WWDG_WR _SFR(uint8_t, WWDG_AddressBase+0x01)

  Window Watchdog Window register (WWDG_WR)
- #define _WWDG_CR_RESET_VALUE ((uint8_t) 0x7F)

  Window Watchdog Control register reset value.
- #define _WWDG_WR_RESET_VALUE ((uint8_t) 0x7F)

  Window Watchdog Window register reset value.
- #define _WWDG_T ((uint8_t) (0x7F << 0))

  Window Watchdog 7-bit counter [6:0] (in _WWDG_CR)
- #define _WWDG_T0 ((uint8_t) (0x01 << 0))

  Window Watchdog 7-bit counter [0] (in _WWDG_CR)
- #define _WWDG_T1 ((uint8_t) (0x01 << 1))

  Window Watchdog 7-bit counter [1] (in _WWDG_CR)
- #define _WWDG_T2 ((uint8_t) (0x01 << 2))

  Window Watchdog 7-bit counter [2] (in _WWDG_CR)
- #define _WWDG_T3 ((uint8_t) (0x01 << 3))

  Window Watchdog 7-bit counter [3] (in _WWDG_CR)
- #define _WWDG_T4 ((uint8_t) (0x01 << 4))

  Window Watchdog 7-bit counter [4] (in _WWDG_CR)
- #define _WWDG_T5 ((uint8_t) (0x01 << 5))

  Window Watchdog 7-bit counter [5] (in _WWDG_CR)
- #define _WWDG_T6 ((uint8_t) (0x01 << 6))

  Window Watchdog 7-bit counter [6] (in _WWDG_CR)
- #define _WWDG_WDGA ((uint8_t) (0x01 << 7))

  Window Watchdog activation bit (n/a if WWDG enabled by option byte) [0] (in _WWDG_CR)
- #define _WWDG_W ((uint8_t) (0x7F << 0))

  Window Watchdog 7-bit window value [6:0] (in _WWDG_WR)

- #define _WWDG_W0 ((uint8_t) (0x01 << 0))
  Window Watchdog 7-bit window value [0] (in _WWDG_WR)
- #define _WWDG_W1 ((uint8_t) (0x01 << 1))
  Window Watchdog 7-bit window value [1] (in _WWDG_WR)
- #define _WWDG_W2 ((uint8_t) (0x01 << 2))
  Window Watchdog 7-bit window value [2] (in _WWDG_WR)
- #define _WWDG_W3 ((uint8_t) (0x01 << 3))
  Window Watchdog 7-bit window value [3] (in _WWDG_WR)
- #define _WWDG_W4 ((uint8_t) (0x01 << 4))
  Window Watchdog 7-bit window value [4] (in _WWDG_WR)
- #define _WWDG_W5 ((uint8_t) (0x01 << 5))
  Window Watchdog 7-bit window value [5] (in _WWDG_WR)
- #define _WWDG_W6 ((uint8_t) (0x01 << 6))
  Window Watchdog 7-bit window value [6] (in _WWDG_WR)
- #define _IWDG _SFR(IWDG_t, IWDG_AddressBase)
  Independent Timeout Watchdog struct/bit access.
- #define _IWDG_KR _SFR(uint8_t, IWDG_AddressBase+0x00)
  Independent Timeout Watchdog Key register (IWDG_KR)
- #define _IWDG_PR _SFR(uint8_t, IWDG_AddressBase+0x01)
  Independent Timeout Watchdog Prescaler register (IWDG_PR)
- #define _IWDG_RLR _SFR(uint8_t, IWDG_AddressBase+0x02)
  Independent Timeout Watchdog Reload register (IWDG_RLR)
- #define _IWDG_PR_RESET_VALUE ((uint8_t) 0x00)
  Independent Timeout Watchdog Prescaler register reset value.
- #define _IWDG_RLR_RESET_VALUE ((uint8_t) 0xFF)
  Independent Timeout Watchdog Reload register reset value.
- #define _IWDG_KEY_ENABLE ((uint8_t) 0xCC)
  Independent Timeout Watchdog enable (in _IWDG_KR)
- #define _IWDG_KEY_REFRESH ((uint8_t) 0xAA)
  Independent Timeout Watchdog refresh (in _IWDG_KR)
- #define _IWDG_KEY_ACCESS ((uint8_t) 0x55)
  Independent Timeout Watchdog unlock write to IWDG_PR and IWDG_RLR (in _IWDG_KR)
- #define _IWDG_PRE ((uint8_t) (0x07 << 0))
  Independent Timeout Watchdog Prescaler divider [2:0] (in _IWDG_PR)
- #define _IWDG_PRE0 ((uint8_t) (0x01 << 0))
  Independent Timeout Watchdog Prescaler divider [0] (in _IWDG_PR)
- #define _IWDG_PRE1 ((uint8_t) (0x01 << 1))
  Independent Timeout Watchdog Prescaler divider [1] (in _IWDG_PR)
- #define _IWDG_PRE2 ((uint8_t) (0x01 << 2))
  Independent Timeout Watchdog Prescaler divider [2] (in _IWDG_PR)
- #define _AWU _SFR(AWU_t, AWU_AddressBase)
  Auto Wake-Up struct/bit access.
- #define _AWU_CSR _SFR(uint8_t, AWU_AddressBase+0x00)
  Auto Wake-Up Control/status register (AWU_CSR)
- #define _AWU_APR _SFR(uint8_t, AWU_AddressBase+0x01)
  Auto Wake-Up Asynchronous prescaler register (AWU_APR)
- #define _AWU_TBR _SFR(uint8_t, AWU_AddressBase+0x02)
  Auto Wake-Up Timebase selection register (AWU_TBR)
- #define _AWU_CSR_RESET_VALUE ((uint8_t) 0x00)
  Auto Wake-Up Control/status register reset value.
- #define _AWU_APR_RESET_VALUE ((uint8_t) 0x3F)

Auto Wake-Up Asynchronous prescaler register reset value.

- #define _AWU_TBR_RESET_VALUE ((uint8_t) 0x00)

  Auto Wake-Up Timebase selection register reset value.
- #define _AWU_MSR ((uint8_t) (0x01 << 0))

  Auto Wake-Up LSI measurement enable [0] (in _AWU_CSR)
- #define _AWU_AWUEN ((uint8_t) (0x01 << 4))

  Auto-wakeup enable [0] (in _AWU_CSR)
- #define _AWU_AWUF ((uint8_t) (0x01 << 5))

  Auto-wakeup status flag [0] (in _AWU_CSR)
- #define _AWU_APRE ((uint8_t) (0x3F << 0))

  Auto-wakeup asynchronous prescaler divider [5:0] (in _AWU_APR)
- #define _AWU_APRE0 ((uint8_t) (0x01 << 0))

  Auto-wakeup asynchronous prescaler divider [0] (in _AWU_APR)
- #define _AWU_APRE1 ((uint8_t) (0x01 << 1))

  Auto-wakeup asynchronous prescaler divider [1] (in _AWU_APR)
- #define _AWU_APRE2 ((uint8_t) (0x01 << 2))

  Auto-wakeup asynchronous prescaler divider [2] (in _AWU_APR)
- #define _AWU_APRE3 ((uint8_t) (0x01 << 3))

  Auto-wakeup asynchronous prescaler divider [3] (in _AWU_APR)
- #define _AWU_APRE4 ((uint8_t) (0x01 << 4))

  Auto-wakeup asynchronous prescaler divider [4] (in _AWU_APR)
- #define _AWU_APRE5 ((uint8_t) (0x01 << 5))

  Auto-wakeup asynchronous prescaler divider [5] (in _AWU_APR)
- #define _AWU_AWUTB ((uint8_t) (0x0F << 0))

  Auto-wakeup timebase selection [3:0] (in _AWU_APR)
- #define _AWU_AWUTB0 ((uint8_t) (0x01 << 0))

  Auto-wakeup timebase selection [0] (in _AWU_APR)
- #define _AWU_AWUTB1 ((uint8_t) (0x01 << 1))

  Auto-wakeup timebase selection [1] (in _AWU_APR)
- #define _AWU_AWUTB2 ((uint8_t) (0x01 << 2))

  Auto-wakeup timebase selection [2] (in _AWU_APR)
- #define _AWU_AWUTB3 ((uint8_t) (0x01 << 3))

  Auto-wakeup timebase selection [3] (in _AWU_APR)
- #define _BEEP _SFR(BEEP_t, BEEP_AddressBase)

  Beeper struct/bit access.
- #define _BEEP_CSR _SFR(uint8_t, BEEP_AddressBase+0x00)

  Beeper control/status register (BEEP_CSR)
- #define _BEEP_CSR_RESET_VALUE ((uint8_t) 0x1F)

  Beeper control/status register reset value.
- #define _BEEP_BEEPDIV ((uint8_t) (0x1F << 0))

  Beeper clock prescaler divider [4:0] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV0 ((uint8_t) (0x01 << 0))

  Beeper clock prescaler divider [0] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV1 ((uint8_t) (0x01 << 1))

  Beeper clock prescaler divider [1] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV2 ((uint8_t) (0x01 << 2))

  Beeper clock prescaler divider [2] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV3 ((uint8_t) (0x01 << 3))

  Beeper clock prescaler divider [3] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV4 ((uint8_t) (0x01 << 4))

  Beeper clock prescaler divider [4] (in _BEEP_CSR)

- #define _BEEP_BEEPEN ((uint8_t) (0x01 << 5))
  Beeper enable [0] (in _BEEP_CSR)
- #define _BEEP_BEEPSEL ((uint8_t) (0x03 << 6))
  Beeper frequency selection [1:0] (in _BEEP_CSR)
- #define _BEEP_BEEPSEL0 ((uint8_t) (0x01 << 6))
  Beeper frequency selection [0] (in _BEEP_CSR)
- #define _BEEP_BEEPSEL1 ((uint8_t) (0x01 << 7))
  Beeper frequency selection [1] (in _BEEP_CSR)
- #define _SPI _SFR(SPI_t, SPI_AddressBase)
  register for SPI control
- #define _SPI_CR1 _SFR(uint8_t, SPI_AddressBase+0x00)
  SPI control register 1.
- #define _SPI_CR2 _SFR(uint8_t, SPI_AddressBase+0x01)
  SPI control register 2.
- #define _SPI_ICR _SFR(uint8_t, SPI_AddressBase+0x02)
  SPI interrupt control register.
- #define _SPI_SR _SFR(uint8_t, SPI_AddressBase+0x03)
  SPI status register.
- #define _SPI_DR _SFR(uint8_t, SPI_AddressBase+0x04)
  SPI data register.
- #define _SPI_CRCPR _SFR(uint8_t, SPI_AddressBase+0x05)
  SPI CRC polynomial register.
- #define _SPI_RXCRCR _SFR(uint8_t, SPI_AddressBase+0x06)
  SPI Rx CRC register.
- #define _SPI_TXCRCR _SFR(uint8_t, SPI_AddressBase+0x07)
  SPI Tx CRC register.
- #define _SPI_CR1_RESET_VALUE ((uint8_t) 0x00)
  SPI Control Register 1 reset value.
- #define _SPI_CR2_RESET_VALUE ((uint8_t) 0x00)
  SPI Control Register 2 reset value.
- #define _SPI_ICR_RESET_VALUE ((uint8_t) 0x00)
  SPI Interrupt Control Register reset value.
- #define _SPI_SR_RESET_VALUE ((uint8_t) 0x02)
  SPI Status Register reset value.
- #define _SPI_DR_RESET_VALUE ((uint8_t) 0x00)
  SPI Data Register reset value.
- #define _SPI_CRCPR_RESET_VALUE ((uint8_t) 0x07)
  SPI Polynomial Register reset value.
- #define _SPI_RXCRCR_RESET_VALUE ((uint8_t) 0x00)
  SPI RX CRC Register reset value.
- #define _SPI_TXCRCR_RESET_VALUE ((uint8_t) 0x00)
  SPI TX CRC Register reset value.
- #define _SPI_CPHA ((uint8_t) (0x01 << 0))
  SPI Clock phase [0] (in _SPI_CR1)
- #define _SPI_CPOL ((uint8_t) (0x01 << 1))
  SPI Clock polarity [0] (in _SPI_CR1)
- #define _SPI_MSTR ((uint8_t) (0x01 << 2))
  SPI Master/slave selection [0] (in _SPI_CR1)
- #define _SPI_BR ((uint8_t) (0x07 << 3))
  SPI Baudrate control [2:0] (in _SPI_CR1)
- #define _SPI_BR0 ((uint8_t) (0x01 << 3))

SPI Baudrate control [0] (in _SPI_CR1)

- #define _SPI_BR1 ((uint8_t) (0x01 << 4))

  SPI Baudrate control [1] (in _SPI_CR1)
- #define _SPI_BR2 ((uint8_t) (0x01 << 5))

  SPI Baudrate control [2] (in _SPI_CR1)
- #define _SPI_SPE ((uint8_t) (0x01 << 6))

  SPI enable [0] (in _SPI_CR1)
- #define _SPI_LSBFIRST ((uint8_t) (0x01 << 7))

  SPI Frame format [0] (in _SPI_CR1)
- #define _SPI_SSI ((uint8_t) (0x01 << 0))

  SPI Internal slave select [0] (in _SPI_CR2)
- #define _SPI_SSM ((uint8_t) (0x01 << 1))

  SPI Software slave management [0] (in _SPI_CR2)
- #define _SPI_RXONLY ((uint8_t) (0x01 << 2))

  SPI Receive only [0] (in _SPI_CR2)
- #define _SPI_CRCNEXT ((uint8_t) (0x01 << 4))

  SPI Transmit CRC next [0] (in _SPI_CR2)
- #define _SPI_CRCEN ((uint8_t) (0x01 << 5))

  SPI Hardware CRC calculation enable [0] (in _SPI_CR2)
- #define _SPI_BDOE ((uint8_t) (0x01 << 6))

  SPI Input/Output enable in bidirectional mode [0] (in _SPI_CR2)
- #define _SPI_BDM ((uint8_t) (0x01 << 7))

  SPI Bidirectional data mode enable [0] (in _SPI_CR2)
- #define _SPI_WKIE ((uint8_t) (0x01 << 4))

  SPI Wakeup interrupt enable [0] (in _SPI_ICR)
- #define _SPI_ERRIE ((uint8_t) (0x01 << 5))

  SPI Error interrupt enable [0] (in _SPI_ICR)
- #define _SPI_RXIE ((uint8_t) (0x01 << 6))

  SPI Rx buffer not empty interrupt enable [0] (in _SPI_ICR)
- #define _SPI_TXIE ((uint8_t) (0x01 << 7))

  SPI Tx buffer empty interrupt enable [0] (in _SPI_ICR)
- #define _SPI_RXNE ((uint8_t) (0x01 << 0))

  SPI Receive buffer not empty [0] (in _SPI_SR)
- #define _SPI_TXE ((uint8_t) (0x01 << 1))

  SPI Transmit buffer empty [0] (in _SPI_SR)
- #define _SPI_WKUP ((uint8_t) (0x01 << 3))

  SPI Wakeup flag [0] (in _SPI_SR)
- #define _SPI_CRCERR ((uint8_t) (0x01 << 4))

  SPI CRC error flag [0] (in _SPI_SR)
- #define _SPI_MODF ((uint8_t) (0x01 << 5))

  SPI Mode fault [0] (in _SPI_SR)
- #define _SPI_OVR ((uint8_t) (0x01 << 6))

  SPI Overrun flag [0] (in _SPI_SR)
- #define _SPI_BSY ((uint8_t) (0x01 << 7))

  SPI Busy flag [0] (in _SPI_SR)
- #define _I2C _SFR(I2C_t, I2C_AddressBase)

  register for SPI control
- #define _I2C_CR1 _SFR(uint8_t, I2C_AddressBase+0x00)

  I2C Control register 1.
- #define _I2C_CR2 _SFR(uint8_t, I2C_AddressBase+0x01)

  I2C Control register 2.

- #define _I2C_FREQR _SFR(uint8_t, I2C_AddressBase+0x02)
  I2C Frequency register.
- #define _I2C_OARL _SFR(uint8_t, I2C_AddressBase+0x03)
  I2C own address register low byte.
- #define _I2C_OARH _SFR(uint8_t, I2C_AddressBase+0x04)
  I2C own address register high byte.
- #define _I2C_DR _SFR(uint8_t, I2C_AddressBase+0x06)
  I2C data register.
- #define _I2C_SR1 _SFR(uint8_t, I2C_AddressBase+0x07)
  I2C Status register 1.
- #define _I2C_SR2 _SFR(uint8_t, I2C_AddressBase+0x08)
  I2C Status register 2.
- #define _I2C_SR3 _SFR(uint8_t, I2C_AddressBase+0x09)
  I2C Status register 3.
- #define _I2C_ITR _SFR(uint8_t, I2C_AddressBase+0x0A)
  I2C Interrupt register.
- #define _I2C_CCRL _SFR(uint8_t, I2C_AddressBase+0x0B)
  I2C Clock control register low byte.
- #define _I2C_CCRH _SFR(uint8_t, I2C_AddressBase+0x0C)
  I2C Clock control register high byte.
- #define _I2C_TRISER _SFR(uint8_t, I2C_AddressBase+0x0D)
  I2C rise time register.
- #define _I2C_CR1_RESET_VALUE ((uint8_t) 0x00)
  I2C Control register 1 reset value.
- #define _I2C_CR2_RESET_VALUE ((uint8_t) 0x00)
  I2C Control register 2 reset value.
- #define _I2C_FREQR_RESET_VALUE ((uint8_t) 0x00)
  I2C Frequency register reset value.
- #define _I2C_OARL_RESET_VALUE ((uint8_t) 0x00)
  I2C own address register low byte reset value.
- #define _I2C_OARH_RESET_VALUE ((uint8_t) 0x00)
  I2C own address register high byte reset value.
- #define _I2C_DR_RESET_VALUE ((uint8_t) 0x00)
  I2C data register reset value.
- #define _I2C_SR1_RESET_VALUE ((uint8_t) 0x00)
  I2C Status register 1 reset value.
- #define _I2C_SR2_RESET_VALUE ((uint8_t) 0x00)
  I2C Status register 2 reset value.
- #define _I2C_SR3_RESET_VALUE ((uint8_t) 0x00)
  I2C Status register 3 reset value.
- #define _I2C_ITR_RESET_VALUE ((uint8_t) 0x00)
  I2C Interrupt register reset value.
- #define _I2C_CCRL_RESET_VALUE ((uint8_t) 0x00)
  I2C Clock control register low byte reset value.
- #define _I2C_CCRH_RESET_VALUE ((uint8_t) 0x00)
  I2C Clock control register high byte reset value.
- #define _I2C_TRISER_RESET_VALUE ((uint8_t) 0x02)
  I2C rise time register reset value.
- #define _I2C_PE ((uint8_t) (0x01 << 0))
  I2C Peripheral enable [0] (in _I2C_CR1)
- #define _I2C_ENGC ((uint8_t) (0x01 << 6))

I2C General call enable [0] (in _I2C_CR1)

- #define _I2C_NOSTRETCH ((uint8_t) (0x01 << 7))

  I2C Clock stretching disable (Slave mode) [0] (in _I2C_CR1)
- #define _I2C_START ((uint8_t) (0x01 << 0))

  I2C Start generation [0] (in _I2C_CR2)
- #define _I2C_STOP ((uint8_t) (0x01 << 1))

  I2C Stop generation [0] (in _I2C_CR2)
- #define _I2C_ACK ((uint8_t) (0x01 << 2))

  I2C Acknowledge enable [0] (in _I2C_CR2)
- #define _I2C_POS ((uint8_t) (0x01 << 3))

  I2C Acknowledge position (for data reception) [0] (in _I2C_CR2)
- #define _I2C_SWRST ((uint8_t) (0x01 << 7))

  I2C Software reset [0] (in _I2C_CR2)
- #define _I2C_FREQ ((uint8_t) (0x3F << 0))

  I2C Peripheral clock frequency [5:0] (in _I2C_FREQR)
- #define _I2C_FREQ0 ((uint8_t) (0x01 << 0))

  I2C Peripheral clock frequency [0] (in _I2C_FREQR)
- #define _I2C_FREQ1 ((uint8_t) (0x01 << 1))

  I2C Peripheral clock frequency [1] (in _I2C_FREQR)
- #define _I2C_FREQ2 ((uint8_t) (0x01 << 2))

  I2C Peripheral clock frequency [2] (in _I2C_FREQR)
- #define _I2C_FREQ3 ((uint8_t) (0x01 << 3))

  I2C Peripheral clock frequency [3] (in _I2C_FREQR)
- #define _I2C_FREQ4 ((uint8_t) (0x01 << 4))

  I2C Peripheral clock frequency [4] (in _I2C_FREQR)
- #define _I2C_FREQ5 ((uint8_t) (0x01 << 5))

  I2C Peripheral clock frequency [5] (in _I2C_FREQR)
- #define _I2C_ADD0 ((uint8_t) (0x01 << 0))

  I2C Interface address [0] (in 10-bit address mode) (in _I2C_OARL)
- #define _I2C_ADD1 ((uint8_t) (0x01 << 1))

  I2C Interface address [1] (in _I2C_OARL)
- #define _I2C_ADD2 ((uint8_t) (0x01 << 2))

  I2C Interface address [2] (in _I2C_OARL)
- #define _I2C_ADD3 ((uint8_t) (0x01 << 3))

  I2C Interface address [3] (in _I2C_OARL)
- #define _I2C_ADD4 ((uint8_t) (0x01 << 4))

  I2C Interface address [4] (in _I2C_OARL)
- #define _I2C_ADD5 ((uint8_t) (0x01 << 5))

  I2C Interface address [5] (in _I2C_OARL)
- #define _I2C_ADD6 ((uint8_t) (0x01 << 6))

  I2C Interface address [6] (in _I2C_OARL)
- #define _I2C_ADD7 ((uint8_t) (0x01 << 7))

  I2C Interface address [7] (in _I2C_OARL)
- #define _I2C_ADD_8_9 ((uint8_t) (0x03 << 1))

  I2C Interface address [9:8] (in 10-bit address mode) (in _I2C_OARH)
- #define _I2C_ADD8 ((uint8_t) (0x01 << 1))

  I2C Interface address [8] (in _I2C_OARH)
- #define _I2C_ADD9 ((uint8_t) (0x01 << 2))

  I2C Interface address [9] (in _I2C_OARH)
- #define _I2C_ADDCONF ((uint8_t) (0x01 << 6))

  I2C Address mode configuration [0] (in _I2C_OARH)

- #define _I2C_ADDMODE ((uint8_t) (0x01 << 7))

  I2C 7-/10-bit addressing mode (Slave mode) [0] (in _I2C_OARH)
- #define _I2C_SB ((uint8_t) (0x01 << 0))

  I2C Start bit (Master mode) [0] (in _I2C_SR1)
- #define _I2C_ADDR ((uint8_t) (0x01 << 1))

  I2C Address sent (Master mode) / matched (Slave mode) [0] (in _I2C_SR1)
- #define _I2C_BTF ((uint8_t) (0x01 << 2))

  I2C Byte transfer finished [0] (in _I2C_SR1)
- #define _I2C_ADD10 ((uint8_t) (0x01 << 3))

  I2C 10-bit header sent (Master mode) [0] (in _I2C_SR1)
- #define _I2C_STOPF ((uint8_t) (0x01 << 4))

  I2C Stop detection (Slave mode) [0] (in _I2C_SR1)
- #define _I2C_RXNE ((uint8_t) (0x01 << 6))

  I2C Data register not empty (receivers) [0] (in _I2C_SR1)
- #define _I2C_TXE ((uint8_t) (0x01 << 7))

  I2C Data register empty (transmitters) [0] (in _I2C_SR1)
- #define _I2C_BERR ((uint8_t) (0x01 << 0))

  I2C Bus error [0] (in _I2C_SR2)
- #define _I2C_ARLO ((uint8_t) (0x01 << 1))

  I2C Arbitration lost (Master mode) [0] (in _I2C_SR2)
- #define _I2C_AF ((uint8_t) (0x01 << 2))

  I2C Acknowledge failure [0] (in _I2C_SR2)
- #define _I2C_OVR ((uint8_t) (0x01 << 3))

  I2C Overrun/underrun [0] (in _I2C_SR2)
- #define _I2C_WUFH ((uint8_t) (0x01 << 5))

  I2C Wakeup from Halt [0] (in _I2C_SR2)
- #define _I2C_MSL ((uint8_t) (0x01 << 0))

  I2C Master/Slave [0] (in _I2C_SR3)
- #define _I2C_BUSY ((uint8_t) (0x01 << 1))

  I2C Bus busy [0] (in _I2C_SR3)
- #define _I2C_TRA ((uint8_t) (0x01 << 2))

  I2C Transmitter/Receiver [0] (in _I2C_SR3)
- #define _I2C_GENCALL ((uint8_t) (0x01 << 4))

  I2C General call header (Slavemode) [0] (in _I2C_SR3)
- #define _I2C_ITERREN ((uint8_t) (0x01 << 0))

  I2C Error interrupt enable [0] (in _I2C_ITR)
- #define _I2C_ITEVTEN ((uint8_t) (0x01 << 1))

  I2C Event interrupt enable [0] (in _I2C_ITR)
- #define _I2C_ITBUFEN ((uint8_t) (0x01 << 2))

  I2C Buffer interrupt enable [0] (in _I2C_ITR)
- #define _I2C_CCR ((uint8_t) (0x0F << 0))

  I2C Clock control register (Master mode) [3:0] (in _I2C_CCRH)
- #define _I2C_CCR0 ((uint8_t) (0x01 << 0))

  I2C Clock control register (Master mode) [0] (in _I2C_CCRH)
- #define _I2C_CCR1 ((uint8_t) (0x01 << 1))

  I2C Clock control register (Master mode) [1] (in _I2C_CCRH)
- #define _I2C_CCR2 ((uint8_t) (0x01 << 2))

  I2C Clock control register (Master mode) [2] (in _I2C_CCRH)
- #define _I2C_CCR3 ((uint8_t) (0x01 << 3))

  I2C Clock control register (Master mode) [3] (in _I2C_CCRH)
- #define _I2C_DUTY ((uint8_t) (0x01 << 6))

I2C Fast mode duty cycle [0] (in _I2C_CCRH)

- #define _I2C_FS ((uint8_t) (0x01 << 7))

  I2C Master mode selection [0] (in _I2C_CCRH)
- #define _I2C_TRISE ((uint8_t) (0x3F << 0))

  I2C Maximum rise time (Master mode) [5:0] (in _I2C_TRISER)
- #define _I2C_TRISE0 ((uint8_t) (0x01 << 0))

  I2C Maximum rise time (Master mode) [0] (in _I2C_TRISER)
- #define _I2C_TRISE1 ((uint8_t) (0x01 << 1))

  I2C Maximum rise time (Master mode) [1] (in _I2C_TRISER)
- #define _I2C_TRISE2 ((uint8_t) (0x01 << 2))

  I2C Maximum rise time (Master mode) [2] (in _I2C_TRISER)
- #define _I2C_TRISE3 ((uint8_t) (0x01 << 3))

  I2C Maximum rise time (Master mode) [3] (in _I2C_TRISER)
- #define _I2C_TRISE4 ((uint8_t) (0x01 << 4))

  I2C Maximum rise time (Master mode) [4] (in _I2C_TRISER)
- #define _I2C_TRISE5 ((uint8_t) (0x01 << 5))

  I2C Maximum rise time (Master mode) [5] (in _I2C_TRISER)
- #define _UART1 _SFR(UART1_t, UART1_AddressBase)

  UART1 struct/bit access.
- #define _UART1_SR _SFR(uint8_t, UART1_AddressBase+0x00)

  UART1 Status register.
- #define _UART1_DR _SFR(uint8_t, UART1_AddressBase+0x01)

  UART1 data register.
- #define _UART1_BRR1 _SFR(uint8_t, UART1_AddressBase+0x02)

  UART1 Baud rate register 1.
- #define _UART1_BRR2 _SFR(uint8_t, UART1_AddressBase+0x03)

  UART1 Baud rate register 2.
- #define _UART1_CR1 _SFR(uint8_t, UART1_AddressBase+0x04)

  UART1 Control register 1.
- #define _UART1_CR2 _SFR(uint8_t, UART1_AddressBase+0x05)

  UART1 Control register 2.
- #define _UART1_CR3 _SFR(uint8_t, UART1_AddressBase+0x06)

  UART1 Control register 3.
- #define _UART1_CR4 _SFR(uint8_t, UART1_AddressBase+0x07)

  UART1 Control register 4.
- #define _UART1_CR5 _SFR(uint8_t, UART1_AddressBase+0x08)

  UART1 Control register 5.
- #define _UART1_GTR _SFR(uint8_t, UART1_AddressBase+0x09)

  UART1 guard time register.
- #define _UART1_PSCR _SFR(uint8_t, UART1_AddressBase+0x0A)

  UART1 prescaler register.
- #define _UART1_SR_RESET_VALUE ((uint8_t) 0xC0)

  UART1 Status register reset value.
- #define _UART1_BRR1_RESET_VALUE ((uint8_t) 0x00)

  UART1 Baud rate register 1 reset value.
- #define _UART1_BRR2_RESET_VALUE ((uint8_t) 0x00)

  UART1 Baud rate register 2 reset value.
- #define _UART1_CR1_RESET_VALUE ((uint8_t) 0x00)

  UART1 Control register 1 reset value.
- #define _UART1_CR2_RESET_VALUE ((uint8_t) 0x00)

  UART1 Control register 2 reset value.

- #define _UART1_CR3_RESET_VALUE ((uint8_t) 0x00)

  UART1 Control register 3 reset value.
- #define _UART1_CR4_RESET_VALUE ((uint8_t) 0x00)

  UART1 Control register 4 reset value.
- #define _UART1_CR5_RESET_VALUE ((uint8_t) 0x00)

  UART1 Control register 5 reset value.
- #define _UART1_GTR_RESET_VALUE ((uint8_t) 0x00)

  UART1 guard time register reset value.
- #define _UART1_PSCR_RESET_VALUE ((uint8_t) 0x00)

  UART1 prescaler register reset value.
- #define _UART1_PE ((uint8_t) (0x01 << 0))

  UART1 Parity error [0] (in _UART1_SR)
- #define _UART1_FE ((uint8_t) (0x01 << 1))

  UART1 Framing error [0] (in _UART1_SR)
- #define _UART1_NF ((uint8_t) (0x01 << 2))

  UART1 Noise flag [0] (in _UART1_SR)
- #define _UART1_OR_LHE ((uint8_t) (0x01 << 3))

  UART1 LIN Header Error (LIN slave mode) / Overrun error [0] (in _UART1_SR)
- #define _UART1_IDLE ((uint8_t) (0x01 << 4))

  UART1 IDLE line detected [0] (in _UART1_SR)
- #define _UART1_RXNE ((uint8_t) (0x01 << 5))

  UART1 Read data register not empty [0] (in _UART1_SR)
- #define _UART1_TC ((uint8_t) (0x01 << 6))

  UART1 Transmission complete [0] (in _UART1_SR)
- #define _UART1_TXE ((uint8_t) (0x01 << 7))

  UART1 Transmit data register empty [0] (in _UART1_SR)
- #define _UART1_PIEN ((uint8_t) (0x01 << 0))

  UART1 Parity interrupt enable [0] (in _UART1_CR1)
- #define _UART1_PS ((uint8_t) (0x01 << 1))

  UART1 Parity selection [0] (in _UART1_CR1)
- #define _UART1_PCEN ((uint8_t) (0x01 << 2))

  UART1 Parity control enable [0] (in _UART1_CR1)
- #define _UART1_WAKE ((uint8_t) (0x01 << 3))

  UART1 Wakeup method [0] (in _UART1_CR1)
- #define _UART1_M ((uint8_t) (0x01 << 4))

  UART1 word length [0] (in _UART1_CR1)
- #define _UART1_UARTD ((uint8_t) (0x01 << 5))

  UART1 Disable (for low power consumption) [0] (in _UART1_CR1)
- #define _UART1_T8 ((uint8_t) (0x01 << 6))

  UART1 Transmit Data bit 8 (in 9-bit mode) [0] (in _UART1_CR1)
- #define _UART1_R8 ((uint8_t) (0x01 << 7))

  UART1 Receive Data bit 8 (in 9-bit mode) [0] (in _UART1_CR1)
- #define _UART1_SBK ((uint8_t) (0x01 << 0))

  UART1 Send break [0] (in _UART1_CR2)
- #define _UART1_RWU ((uint8_t) (0x01 << 1))

  UART1 Receiver wakeup [0] (in _UART1_CR2)
- #define _UART1_REN ((uint8_t) (0x01 << 2))

  UART1 Receiver enable [0] (in _UART1_CR2)
- #define _UART1_TEN ((uint8_t) (0x01 << 3))

  UART1 Transmitter enable [0] (in _UART1_CR2)
- #define _UART1_ILIEN ((uint8_t) (0x01 << 4))

UART1 IDLE Line interrupt enable [0] (in _UART1_CR2)

- #define _UART1_RIEN ((uint8_t) (0x01 << 5))

  UART1 Receiver interrupt enable [0] (in _UART1_CR2)
- #define _UART1_TCIEN ((uint8_t) (0x01 << 6))

  UART1 Transmission complete interrupt enable [0] (in _UART1_CR2)
- #define _UART1_TIEN ((uint8_t) (0x01 << 7))

  UART1 Transmitter interrupt enable [0] (in _UART1_CR2)
- #define _UART1_LBCL ((uint8_t) (0x01 << 0))

  UART1 Last bit clock pulse [0] (in _UART1_CR3)
- #define _UART1_CPHA ((uint8_t) (0x01 << 1))

  UART1 Clock phase [0] (in _UART1_CR3)
- #define _UART1_CPOL ((uint8_t) (0x01 << 2))

  UART1 Clock polarity [0] (in _UART1_CR3)
- #define _UART1_CKEN ((uint8_t) (0x01 << 3))

  UART1 Clock enable [0] (in _UART1_CR3)
- #define _UART1_STOP ((uint8_t) (0x03 << 4))

  UART1 STOP bits [1:0] (in _UART1_CR3)
- #define _UART1_STOP0 ((uint8_t) (0x01 << 4))

  UART1 STOP bits [0] (in _UART1_CR3)
- #define _UART1_STOP1 ((uint8_t) (0x01 << 5))

  UART1 STOP bits [1] (in _UART1_CR3)
- #define _UART1_LINEN ((uint8_t) (0x01 << 6))

  UART1 LIN mode enable [0] (in _UART1_CR3)
- #define _UART1_ADD ((uint8_t) (0x0F << 0))

  UART1 Address of the UART node [3:0] (in _UART1_CR4)
- #define _UART1_ADD0 ((uint8_t) (0x01 << 0))

  UART1 Address of the UART node [0] (in _UART1_CR4)
- #define _UART1_ADD1 ((uint8_t) (0x01 << 1))

  UART1 Address of the UART node [1] (in _UART1_CR4)
- #define _UART1_ADD2 ((uint8_t) (0x01 << 2))

  UART1 Address of the UART node [2] (in _UART1_CR4)
- #define _UART1_ADD3 ((uint8_t) (0x01 << 3))

  UART1 Address of the UART node [3] (in _UART1_CR4)
- #define _UART1_LBDF ((uint8_t) (0x01 << 4))

  UART1 LIN Break Detection Flag [0] (in _UART1_CR4)
- #define _UART1_LBDL ((uint8_t) (0x01 << 5))

  UART1 LIN Break Detection Length [0] (in _UART1_CR4)
- #define _UART1_LBDIEN ((uint8_t) (0x01 << 6))

  UART1 LIN Break Detection Interrupt Enable [0] (in _UART1_CR4)
- #define _UART1_IREN ((uint8_t) (0x01 << 1))

  UART1 IrDA mode Enable [0] (in _UART1_CR5)
- #define _UART1_IRLP ((uint8_t) (0x01 << 2))

  UART1 IrDA Low Power [0] (in _UART1_CR5)
- #define _UART1_HDSEL ((uint8_t) (0x01 << 3))

  UART1 Half-Duplex Selection [0] (in _UART1_CR5)
- #define _UART1_NACK ((uint8_t) (0x01 << 4))

  UART1 Smartcard NACK enable [0] (in _UART1_CR5)
- #define _UART1_SCEN ((uint8_t) (0x01 << 5))

  UART1 Smartcard mode enable [0] (in _UART1_CR5)
- #define _UART2 _SFR(UART2_t, UART2_AddressBase)

  UART2 struct/bit access.

- #define _UART2_SR _SFR(uint8_t, UART2_AddressBase+0x00)
  UART2 Status register.
- #define _UART2_DR _SFR(uint8_t, UART2_AddressBase+0x01)
  UART2 data register.
- #define _UART2_BRR1 _SFR(uint8_t, UART2_AddressBase+0x02)
  UART2 Baud rate register 1.
- #define _UART2_BRR2 _SFR(uint8_t, UART2_AddressBase+0x03)
  UART2 Baud rate register 2.
- #define _UART2_CR1 _SFR(uint8_t, UART2_AddressBase+0x04)
  UART2 Control register 1.
- #define _UART2_CR2 _SFR(uint8_t, UART2_AddressBase+0x05)
  UART2 Control register 2.
- #define _UART2_CR3 _SFR(uint8_t, UART2_AddressBase+0x06)
  UART2 Control register 3.
- #define _UART2_CR4 _SFR(uint8_t, UART2_AddressBase+0x07)
  UART2 Control register 4.
- #define _UART2_CR5 _SFR(uint8_t, UART2_AddressBase+0x08)
  UART2 Control register 5.
- #define _UART2_CR6 _SFR(uint8_t, UART2_AddressBase+0x09)
  UART2 Control register 6.
- #define _UART2_GTR _SFR(uint8_t, UART2_AddressBase+0x0A)
  UART2 guard time register.
- #define _UART2_PSCR _SFR(uint8_t, UART2_AddressBase+0x0B)
  UART2 prescaler register.
- #define _UART2_SR_RESET_VALUE ((uint8_t) 0xC0)
  UART2 Status register reset value.
- #define _UART2_BRR1_RESET_VALUE ((uint8_t) 0x00)
  UART2 Baud rate register 1 reset value.
- #define _UART2_BRR2_RESET_VALUE ((uint8_t) 0x00)
  UART2 Baud rate register 2 reset value.
- #define _UART2_CR1_RESET_VALUE ((uint8_t) 0x00)
  UART2 Control register 1 reset value.
- #define _UART2_CR2_RESET_VALUE ((uint8_t) 0x00)
  UART2 Control register 2 reset value.
- #define _UART2_CR3_RESET_VALUE ((uint8_t) 0x00)
  UART2 Control register 3 reset value.
- #define _UART2_CR4_RESET_VALUE ((uint8_t) 0x00)
  UART2 Control register 4 reset value.
- #define _UART2_CR5_RESET_VALUE ((uint8_t) 0x00)
  UART2 Control register 5 reset value.
- #define _UART2_CR6_RESET_VALUE ((uint8_t) 0x00)
  UART2 Control register 6 reset value.
- #define _UART2_GTR_RESET_VALUE ((uint8_t) 0x00)
  UART2 guard time register reset value.
- #define _UART2_PSCR_RESET_VALUE ((uint8_t) 0x00)
  UART2 prescaler register reset value.
- #define _UART2_PE ((uint8_t) (0x01 << 0))
  UART2 Parity error [0] (in _UART2_SR)
- #define _UART2_FE ((uint8_t) (0x01 << 1))
  UART2 Framing error [0] (in _UART2_SR)
- #define _UART2_NF ((uint8_t) (0x01 << 2))

UART2 Noise flag [0] (in _UART2_SR)

- #define _UART2_OR_LHE ((uint8_t) (0x01 << 3))

  UART2 LIN Header Error (LIN slave mode) / Overrun error [0] (in _UART2_SR)
- #define _UART2_IDLE ((uint8_t) (0x01 << 4))

  UART2 IDLE line detected [0] (in _UART2_SR)
- #define _UART2_RXNE ((uint8_t) (0x01 << 5))

  UART2 Read data register not empty [0] (in _UART2_SR)
- #define _UART2_TC ((uint8_t) (0x01 << 6))

  UART2 Transmission complete [0] (in _UART2_SR)
- #define _UART2_TXE ((uint8_t) (0x01 << 7))

  UART2 Transmit data register empty [0] (in _UART2_SR)
- #define _UART2_PIEN ((uint8_t) (0x01 << 0))

  UART2 Parity interrupt enable [0] (in _UART2_CR1)
- #define _UART2_PS ((uint8_t) (0x01 << 1))

  UART2 Parity selection [0] (in _UART2_CR1)
- #define _UART2_PCEN ((uint8_t) (0x01 << 2))

  UART2 Parity control enable [0] (in _UART2_CR1)
- #define _UART2_WAKE ((uint8_t) (0x01 << 3))

  UART2 Wakeup method [0] (in _UART2_CR1)
- #define _UART2_M ((uint8_t) (0x01 << 4))

  UART2 word length [0] (in _UART2_CR1)
- #define _UART2_UARTD ((uint8_t) (0x01 << 5))

  UART2 Disable (for low power consumption) [0] (in _UART2_CR1)
- #define _UART2_T8 ((uint8_t) (0x01 << 6))

  UART2 Transmit Data bit 8 (in 9-bit mode) [0] (in _UART2_CR1)
- #define _UART2_R8 ((uint8_t) (0x01 << 7))

  UART2 Receive Data bit 8 (in 9-bit mode) [0] (in _UART2_CR1)
- #define _UART2_SBK ((uint8_t) (0x01 << 0))

  UART2 Send break [0] (in _UART2_CR2)
- #define _UART2_RWU ((uint8_t) (0x01 << 1))

  UART2 Receiver wakeup [0] (in _UART2_CR2)
- #define _UART2_REN ((uint8_t) (0x01 << 2))

  UART2 Receiver enable [0] (in _UART2_CR2)
- #define _UART2_TEN ((uint8_t) (0x01 << 3))

  UART2 Transmitter enable [0] (in _UART2_CR2)
- #define _UART2_ILIEN ((uint8_t) (0x01 << 4))

  UART2 IDLE Line interrupt enable [0] (in _UART2_CR2)
- #define _UART2_RIEN ((uint8_t) (0x01 << 5))

  UART2 Receiver interrupt enable [0] (in _UART2_CR2)
- #define _UART2_TCIEN ((uint8_t) (0x01 << 6))

  UART2 Transmission complete interrupt enable [0] (in _UART2_CR2)
- #define _UART2_TIEN ((uint8_t) (0x01 << 7))

  UART2 Transmitter interrupt enable [0] (in _UART2_CR2)
- #define _UART2_LBCL ((uint8_t) (0x01 << 0))

  UART2 Last bit clock pulse [0] (in _UART2_CR3)
- #define _UART2_CPHA ((uint8_t) (0x01 << 1))

  UART2 Clock phase [0] (in _UART2_CR3)
- #define _UART2_CPOL ((uint8_t) (0x01 << 2))

  UART2 Clock polarity [0] (in _UART2_CR3)
- #define _UART2_CKEN ((uint8_t) (0x01 << 3))

  UART2 Clock enable [0] (in _UART2_CR3)

- #define _UART2_STOP ((uint8_t) (0x03 << 4))
  UART2 STOP bits [1:0] (in _UART2_CR3)
- #define _UART2_STOP0 ((uint8_t) (0x01 << 4))
  UART2 STOP bits [0] (in _UART2_CR3)
- #define _UART2_STOP1 ((uint8_t) (0x01 << 5))
  UART2 STOP bits [1] (in _UART2_CR3)
- #define _UART2_LINEN ((uint8_t) (0x01 << 6))
  UART2 LIN mode enable [0] (in _UART2_CR3)
- #define _UART2_ADD ((uint8_t) (0x0F << 0))
  UART2 Address of the UART node [3:0] (in _UART2_CR4)
- #define _UART2_ADD0 ((uint8_t) (0x01 << 0))
  UART2 Address of the UART node [0] (in _UART2_CR4)
- #define _UART2_ADD1 ((uint8_t) (0x01 << 1))
  UART2 Address of the UART node [1] (in _UART2_CR4)
- #define _UART2_ADD2 ((uint8_t) (0x01 << 2))
  UART2 Address of the UART node [2] (in _UART2_CR4)
- #define _UART2_ADD3 ((uint8_t) (0x01 << 3))
  UART2 Address of the UART node [3] (in _UART2_CR4)
- #define _UART2_LBDF ((uint8_t) (0x01 << 4))
  UART2 LIN Break Detection Flag [0] (in _UART2_CR4)
- #define _UART2_LBDL ((uint8_t) (0x01 << 5))
  UART2 LIN Break Detection Length [0] (in _UART2_CR4)
- #define _UART2_LBDIEN ((uint8_t) (0x01 << 6))
  UART2 LIN Break Detection Interrupt Enable [0] (in _UART2_CR4)
- #define _UART2_IREN ((uint8_t) (0x01 << 1))
  UART2 IrDA mode Enable [0] (in _UART2_CR5)
- #define _UART2_IRLP ((uint8_t) (0x01 << 2))
  UART2 IrDA Low Power [0] (in _UART2_CR5)
- #define _UART2_NACK ((uint8_t) (0x01 << 4))
  UART2 Smartcard NACK enable [0] (in _UART2_CR5)
- #define _UART2_SCEN ((uint8_t) (0x01 << 5))
  UART2 Smartcard mode enable [0] (in _UART2_CR5)
- #define _UART2_LSF ((uint8_t) (0x01 << 0))
  UART2 LIN Sync Field [0] (in _UART2_CR6)
- #define _UART2_LHDF ((uint8_t) (0x01 << 1))
  UART2 LIN Header Detection Flag [0] (in _UART2_CR6)
- #define _UART2_LHDIEN ((uint8_t) (0x01 << 2))
  UART2 LIN Header Detection Interrupt Enable [0] (in _UART2_CR6)
- #define _UART2_LASE ((uint8_t) (0x01 << 4))
  UART2 LIN automatic resynchronisation enable [0] (in _UART2_CR6)
- #define _UART2_LSLV ((uint8_t) (0x01 << 5))
  UART2 LIN Slave Enable [0] (in _UART2_CR6)
- #define _UART2_LDUM ((uint8_t) (0x01 << 7))
  UART2 LIN Divider Update Method [0] (in _UART2_CR6)
- #define _UART3 _SFR(UART3_t, UART3_AddressBase)
  UART3 struct/bit access.
- #define _UART3_SR _SFR(uint8_t, UART3_AddressBase+0x00)
  UART3 Status register.
- #define _UART3_DR _SFR(uint8_t, UART3_AddressBase+0x01)
  UART3 data register.
- #define _UART3_BRR1 _SFR(uint8_t, UART3_AddressBase+0x02)

UART3 Baud rate register 1.

- #define _UART3_BRR2 _SFR(uint8_t, UART3_AddressBase+0x03)

  UART3 Baud rate register 2.
- #define _UART3_CR1 _SFR(uint8_t, UART3_AddressBase+0x04)

  UART3 Control register 1.
- #define _UART3_CR2 _SFR(uint8_t, UART3_AddressBase+0x05)

  UART3 Control register 2.
- #define _UART3_CR3 _SFR(uint8_t, UART3_AddressBase+0x06)

  UART3 Control register 3.
- #define _UART3_CR4 _SFR(uint8_t, UART3_AddressBase+0x07)

  UART3 Control register 4.
- #define _UART3_CR6 _SFR(uint8_t, UART3_AddressBase+0x09)

  UART3 Control register 6.
- #define _UART3_SR_RESET_VALUE ((uint8_t) 0xC0)

  UART3 Status register reset value.
- #define _UART3_BRR1_RESET_VALUE ((uint8_t) 0x00)

  UART3 Baud rate register 1 reset value.
- #define _UART3_BRR2_RESET_VALUE ((uint8_t) 0x00)

  UART3 Baud rate register 2 reset value.
- #define _UART3_CR1_RESET_VALUE ((uint8_t) 0x00)

  UART3 Control register 1 reset value.
- #define _UART3_CR2_RESET_VALUE ((uint8_t) 0x00)

  UART3 Control register 2 reset value.
- #define _UART3_CR3_RESET_VALUE ((uint8_t) 0x00)

  UART3 Control register 3 reset value.
- #define _UART3_CR4_RESET_VALUE ((uint8_t) 0x00)

  UART3 Control register 4 reset value.
- #define _UART3_CR6_RESET_VALUE ((uint8_t) 0x00)

  UART3 Control register 6 reset value.
- #define _UART3_PE ((uint8_t) (0x01 << 0))

  UART3 Parity error [0] (in _UART3_SR)
- #define _UART3_FE ((uint8_t) (0x01 << 1))

  UART3 Framing error [0] (in _UART3_SR)
- #define _UART3_NF ((uint8_t) (0x01 << 2))

  UART3 Noise flag [0] (in _UART3_SR)
- #define _UART3_OR_LHE ((uint8_t) (0x01 << 3))

  UART3 LIN Header Error (LIN slave mode) / Overrun error [0] (in _UART3_SR)
- #define _UART3_IDLE ((uint8_t) (0x01 << 4))

  UART3 IDLE line detected [0] (in _UART3_SR)
- #define _UART3_RXNE ((uint8_t) (0x01 << 5))

  UART3 Read data register not empty [0] (in _UART3_SR)
- #define _UART3_TC ((uint8_t) (0x01 << 6))

  UART3 Transmission complete [0] (in _UART3_SR)
- #define _UART3_TXE ((uint8_t) (0x01 << 7))

  UART3 Transmit data register empty [0] (in _UART3_SR)
- #define _UART3_PIEN ((uint8_t) (0x01 << 0))

  UART3 Parity interrupt enable [0] (in _UART3_CR1)
- #define _UART3_PS ((uint8_t) (0x01 << 1))

  UART3 Parity selection [0] (in _UART3_CR1)
- #define _UART3_PCEN ((uint8_t) (0x01 << 2))

  UART3 Parity control enable [0] (in _UART3_CR1)

- #define _UART3_WAKE ((uint8_t) (0x01 << 3))
  UART3 Wakeup method [0] (in _UART3_CR1)
- #define _UART3_M ((uint8_t) (0x01 << 4))
  UART3 word length [0] (in _UART3_CR1)
- #define _UART3_UARTD ((uint8_t) (0x01 << 5))
  UART3 Disable (for low power consumption) [0] (in _UART3_CR1)
- #define _UART3_T8 ((uint8_t) (0x01 << 6))
  UART3 Transmit Data bit 8 (in 9-bit mode) [0] (in _UART3_CR1)
- #define _UART3_R8 ((uint8_t) (0x01 << 7))
  UART3 Receive Data bit 8 (in 9-bit mode) [0] (in _UART3_CR1)
- #define _UART3_SBK ((uint8_t) (0x01 << 0))
  UART3 Send break [0] (in _UART3_CR2)
- #define _UART3_RWU ((uint8_t) (0x01 << 1))
  UART3 Receiver wakeup [0] (in _UART3_CR2)
- #define _UART3_REN ((uint8_t) (0x01 << 2))
  UART3 Receiver enable [0] (in _UART3_CR2)
- #define _UART3_TEN ((uint8_t) (0x01 << 3))
  UART3 Transmitter enable [0] (in _UART3_CR2)
- #define _UART3_ILIEN ((uint8_t) (0x01 << 4))
  UART3 IDLE Line interrupt enable [0] (in _UART3_CR2)
- #define _UART3_RIEN ((uint8_t) (0x01 << 5))
  UART3 Receiver interrupt enable [0] (in _UART3_CR2)
- #define _UART3_TCIEN ((uint8_t) (0x01 << 6))
  UART3 Transmission complete interrupt enable [0] (in _UART3_CR2)
- #define _UART3_TIEN ((uint8_t) (0x01 << 7))
  UART3 Transmitter interrupt enable [0] (in _UART3_CR2)
- #define _UART3_STOP ((uint8_t) (0x03 << 4))
  UART3 STOP bits [1:0] (in _UART3_CR3)
- #define _UART3_STOP0 ((uint8_t) (0x01 << 4))
  UART3 STOP bits [0] (in _UART3_CR3)
- #define _UART3_STOP1 ((uint8_t) (0x01 << 5))
  UART3 STOP bits [1] (in _UART3_CR3)
- #define _UART3_LINEN ((uint8_t) (0x01 << 6))
  UART3 LIN mode enable [0] (in _UART3_CR3)
- #define _UART3_ADD ((uint8_t) (0x0F << 0))
  UART3 Address of the UART node [3:0] (in _UART3_CR4)
- #define _UART3_ADD0 ((uint8_t) (0x01 << 0))
  UART3 Address of the UART node [0] (in _UART3_CR4)
- #define _UART3_ADD1 ((uint8_t) (0x01 << 1))
  UART3 Address of the UART node [1] (in _UART3_CR4)
- #define _UART3_ADD2 ((uint8_t) (0x01 << 2))
  UART3 Address of the UART node [2] (in _UART3_CR4)
- #define _UART3_ADD3 ((uint8_t) (0x01 << 3))
  UART3 Address of the UART node [3] (in _UART3_CR4)
- #define _UART3_LBDF ((uint8_t) (0x01 << 4))
  UART3 LIN Break Detection Flag [0] (in _UART3_CR4)
- #define _UART3_LBDL ((uint8_t) (0x01 << 5))
  UART3 LIN Break Detection Length [0] (in _UART3_CR4)
- #define _UART3_LBDIEN ((uint8_t) (0x01 << 6))
  UART3 LIN Break Detection Interrupt Enable [0] (in _UART3_CR4)
- #define _UART3_LSF ((uint8_t) (0x01 << 0))

UART3 LIN Sync Field [0] (in _UART3_CR6)

- #define _UART3_LHDF ((uint8_t) (0x01 << 1))

  UART3 LIN Header Detection Flag [0] (in _UART3_CR6)
- #define _UART3_LHDIEN ((uint8_t) (0x01 << 2))

  UART3 LIN Header Detection Interrupt Enable [0] (in _UART3_CR6)
- #define _UART3_LASE ((uint8_t) (0x01 << 4))

  UART3 LIN automatic resynchronisation enable [0] (in _UART3_CR6)
- #define _UART3_LSLV ((uint8_t) (0x01 << 5))

  UART3 LIN Slave Enable [0] (in _UART3_CR6)
- #define _UART3_LDUM ((uint8_t) (0x01 << 7))

  UART3 LIN Divider Update Method [0] (in _UART3_CR6)
- #define _UART4 _SFR(UART4_t, UART4_AddressBase)

  UART4 struct/bit access.
- #define _UART4_SR _SFR(uint8_t, UART4_AddressBase+0x00)

  UART4 Status register.
- #define _UART4_DR _SFR(uint8_t, UART4_AddressBase+0x01)

  UART4 data register.
- #define _UART4_BRR1 _SFR(uint8_t, UART4_AddressBase+0x02)

  UART4 Baud rate register 1.
- #define _UART4_BRR2 _SFR(uint8_t, UART4_AddressBase+0x03)

  UART4 Baud rate register 2.
- #define _UART4_CR1 _SFR(uint8_t, UART4_AddressBase+0x04)

  UART4 Control register 1.
- #define _UART4_CR2 _SFR(uint8_t, UART4_AddressBase+0x05)

  UART4 Control register 2.
- #define _UART4_CR3 _SFR(uint8_t, UART4_AddressBase+0x06)

  UART4 Control register 3.
- #define _UART4_CR4 _SFR(uint8_t, UART4_AddressBase+0x07)

  UART4 Control register 4.
- #define _UART4_CR5 _SFR(uint8_t, UART4_AddressBase+0x08)

  UART4 Control register 5.
- #define _UART4_CR6 _SFR(uint8_t, UART4_AddressBase+0x09)

  UART4 Control register 6.
- #define _UART4_GTR _SFR(uint8_t, UART4_AddressBase+0x0A)

  UART4 guard time register.
- #define _UART4_PSCR _SFR(uint8_t, UART4_AddressBase+0x0B)

  UART4 prescaler register.
- #define _UART4_SR_RESET_VALUE ((uint8_t) 0xC0)

  UART4 Status register reset value.
- #define _UART4_BRR1_RESET_VALUE ((uint8_t) 0x00)

  UART4 Baud rate register 1 reset value.
- #define _UART4_BRR2_RESET_VALUE ((uint8_t) 0x00)

  UART4 Baud rate register 2 reset value.
- #define _UART4_CR1_RESET_VALUE ((uint8_t) 0x00)

  UART4 Control register 1 reset value.
- #define _UART4_CR2_RESET_VALUE ((uint8_t) 0x00)

  UART4 Control register 2 reset value.
- #define _UART4_CR3_RESET_VALUE ((uint8_t) 0x00)

  UART4 Control register 3 reset value.
- #define _UART4_CR4_RESET_VALUE ((uint8_t) 0x00)

  UART4 Control register 4 reset value.

- #define _UART4_CR5_RESET_VALUE ((uint8_t) 0x00)

  UART4 Control register 5 reset value.
- #define _UART4_CR6_RESET_VALUE ((uint8_t) 0x00)

  UART4 Control register 6 reset value.
- #define _UART4_GTR_RESET_VALUE ((uint8_t) 0x00)

  UART4 guard time register reset value.
- #define _UART4_PSCR_RESET_VALUE ((uint8_t) 0x00)

  UART4 prescaler register reset value.
- #define _UART4_PE ((uint8_t) (0x01 << 0))

  UART4 Parity error [0] (in _UART4_SR)
- #define _UART4_FE ((uint8_t) (0x01 << 1))

  UART4 Framing error [0] (in _UART4_SR)
- #define _UART4_NF ((uint8_t) (0x01 << 2))

  UART4 Noise flag [0] (in _UART4_SR)
- #define _UART4_OR_LHE ((uint8_t) (0x01 << 3))

  UART4 LIN Header Error (LIN slave mode) / Overrun error [0] (in _UART4_SR)
- #define _UART4_IDLE ((uint8_t) (0x01 << 4))

  UART4 IDLE line detected [0] (in _UART4_SR)
- #define _UART4_RXNE ((uint8_t) (0x01 << 5))

  UART4 Read data register not empty [0] (in _UART4_SR)
- #define _UART4_TC ((uint8_t) (0x01 << 6))

  UART4 Transmission complete [0] (in _UART4_SR)
- #define _UART4_TXE ((uint8_t) (0x01 << 7))

  UART4 Transmit data register empty [0] (in _UART4_SR)
- #define _UART4_PIEN ((uint8_t) (0x01 << 0))

  UART4 Parity interrupt enable [0] (in _UART4_CR1)
- #define _UART4_PS ((uint8_t) (0x01 << 1))

  UART4 Parity selection [0] (in _UART4_CR1)
- #define _UART4_PCEN ((uint8_t) (0x01 << 2))

  UART4 Parity control enable [0] (in _UART4_CR1)
- #define _UART4_WAKE ((uint8_t) (0x01 << 3))

  UART4 Wakeup method [0] (in _UART4_CR1)
- #define _UART4_M ((uint8_t) (0x01 << 4))

  UART4 word length [0] (in _UART4_CR1)
- #define _UART4_UARTD ((uint8_t) (0x01 << 5))

  UART4 Disable (for low power consumption) [0] (in _UART4_CR1)
- #define _UART4_T8 ((uint8_t) (0x01 << 6))

  UART4 Transmit Data bit 8 (in 9-bit mode) [0] (in _UART4_CR1)
- #define _UART4_R8 ((uint8_t) (0x01 << 7))

  UART4 Receive Data bit 8 (in 9-bit mode) [0] (in _UART4_CR1)
- #define _UART4_SBK ((uint8_t) (0x01 << 0))

  UART4 Send break [0] (in _UART4_CR2)
- #define _UART4_RWU ((uint8_t) (0x01 << 1))

  UART4 Receiver wakeup [0] (in _UART4_CR2)
- #define _UART4_REN ((uint8_t) (0x01 << 2))

  UART4 Receiver enable [0] (in _UART4_CR2)
- #define _UART4_TEN ((uint8_t) (0x01 << 3))

  UART4 Transmitter enable [0] (in _UART4_CR2)
- #define _UART4_ILIEN ((uint8_t) (0x01 << 4))

  UART4 IDLE Line interrupt enable [0] (in _UART4_CR2)
- #define _UART4_RIEN ((uint8_t) (0x01 << 5))

UART4 Receiver interrupt enable [0] (in _UART4_CR2)

- #define _UART4_TCIEN ((uint8_t) (0x01 << 6))

  UART4 Transmission complete interrupt enable [0] (in _UART4_CR2)
- #define _UART4_TIEN ((uint8_t) (0x01 << 7))

  UART4 Transmitter interrupt enable [0] (in _UART4_CR2)
- #define _UART4_LBCL ((uint8_t) (0x01 << 0))

  UART4 Last bit clock pulse [0] (in _UART4_CR3)
- #define _UART4_CPHA ((uint8_t) (0x01 << 1))

  UART4 Clock phase [0] (in _UART4_CR3)
- #define _UART4_CPOL ((uint8_t) (0x01 << 2))

  UART4 Clock polarity [0] (in _UART4_CR3)
- #define _UART4_CKEN ((uint8_t) (0x01 << 3))

  UART4 Clock enable [0] (in _UART4_CR3)
- #define _UART4_STOP ((uint8_t) (0x03 << 4))

  UART4 STOP bits [1:0] (in _UART4_CR3)
- #define _UART4_STOP0 ((uint8_t) (0x01 << 4))

  UART4 STOP bits [0] (in _UART4_CR3)
- #define _UART4_STOP1 ((uint8_t) (0x01 << 5))

  UART4 STOP bits [1] (in _UART4_CR3)
- #define _UART4_LINEN ((uint8_t) (0x01 << 6))

  UART4 LIN mode enable [0] (in _UART4_CR3)
- #define _UART4_ADD ((uint8_t) (0x0F << 0))

  UART4 Address of the UART node [3:0] (in _UART4_CR4)
- #define _UART4_ADD0 ((uint8_t) (0x01 << 0))

  UART4 Address of the UART node [0] (in _UART4_CR4)
- #define _UART4_ADD1 ((uint8_t) (0x01 << 1))

  UART4 Address of the UART node [1] (in _UART4_CR4)
- #define _UART4_ADD2 ((uint8_t) (0x01 << 2))

  UART4 Address of the UART node [2] (in _UART4_CR4)
- #define _UART4_ADD3 ((uint8_t) (0x01 << 3))

  UART4 Address of the UART node [3] (in _UART4_CR4)
- #define _UART4_LBDF ((uint8_t) (0x01 << 4))

  UART4 LIN Break Detection Flag [0] (in _UART4_CR4)
- #define _UART4_LBDL ((uint8_t) (0x01 << 5))

  UART4 LIN Break Detection Length [0] (in _UART4_CR4)
- #define _UART4_LBDIEN ((uint8_t) (0x01 << 6))

  UART4 LIN Break Detection Interrupt Enable [0] (in _UART4_CR4)
- #define _UART4_IREN ((uint8_t) (0x01 << 1))

  UART4 IrDA mode Enable [0] (in _UART4_CR5)
- #define _UART4_IRLP ((uint8_t) (0x01 << 2))

  UART4 IrDA Low Power [0] (in _UART4_CR5)
- #define _UART4_HDSEL ((uint8_t) (0x01 << 3))

  UART4 Half-Duplex Selection [0] (in _UART4_CR5)
- #define _UART4_NACK ((uint8_t) (0x01 << 4))

  UART4 Smartcard NACK enable [0] (in _UART4_CR5)
- #define _UART4_SCEN ((uint8_t) (0x01 << 5))

  UART4 Smartcard mode enable [0] (in _UART4_CR5)
- #define _UART4_LSF ((uint8_t) (0x01 << 0))

  UART4 LIN Sync Field [0] (in _UART4_CR6)
- #define _UART4_LHDF ((uint8_t) (0x01 << 1))

  UART4 LIN Header Detection Flag [0] (in _UART4_CR6)

- #define _UART4_LHDIEN ((uint8_t) (0x01 << 2))

  UART4 LIN Header Detection Interrupt Enable [0] (in _UART4_CR6)
- #define _UART4_LASE ((uint8_t) (0x01 << 4))

  UART4 LIN automatic resynchronisation enable [0] (in _UART4_CR6)
- #define _UART4_LSLV ((uint8_t) (0x01 << 5))

  UART4 LIN Slave Enable [0] (in _UART4_CR6)
- #define _UART4_LDUM ((uint8_t) (0x01 << 7))

  UART4 LIN Divider Update Method [0] (in _UART4_CR6)
- #define _TIM1 _SFR(TIM1_t, TIM1_AddressBase)

  TIM1 struct/bit access.
- #define _TIM1_CR1 _SFR(uint8_t, TIM1_AddressBase+0x00)

  TIM1 control register 1.
- #define _TIM1_CR2 _SFR(uint8_t, TIM1_AddressBase+0x01)

  TIM1 control register 2.
- #define _TIM1_SMCR _SFR(uint8_t, TIM1_AddressBase+0x02)

  TIM1 Slave mode control register.
- #define _TIM1_ETR _SFR(uint8_t, TIM1_AddressBase+0x03)

  TIM1 External trigger register.
- #define _TIM1_IER _SFR(uint8_t, TIM1_AddressBase+0x04)

  TIM1 interrupt enable register.
- #define _TIM1_SR1 _SFR(uint8_t, TIM1_AddressBase+0x05)

  TIM1 status register 1.
- #define _TIM1_SR2 _SFR(uint8_t, TIM1_AddressBase+0x06)

  TIM1 status register 2.
- #define _TIM1_EGR _SFR(uint8_t, TIM1_AddressBase+0x07)

  TIM1 Event generation register.
- #define _TIM1_CCMR1 _SFR(uint8_t, TIM1_AddressBase+0x08)

  TIM1 Capture/compare mode register 1.
- #define _TIM1_CCMR2 _SFR(uint8_t, TIM1_AddressBase+0x09)

  TIM1 Capture/compare mode register 2.
- #define _TIM1_CCMR3 _SFR(uint8_t, TIM1_AddressBase+0x0A)

  TIM1 Capture/compare mode register 3.
- #define _TIM1_CCMR4 _SFR(uint8_t, TIM1_AddressBase+0x0B)

  TIM1 Capture/compare mode register 4.
- #define _TIM1_CCER1 _SFR(uint8_t, TIM1_AddressBase+0x0C)

  TIM1 Capture/compare enable register 1.
- #define _TIM1_CCER2 _SFR(uint8_t, TIM1_AddressBase+0x0D)

  TIM1 Capture/compare enable register 2.
- #define _TIM1_CNTRH _SFR(uint8_t, TIM1_AddressBase+0x0E)

  TIM1 counter register high byte.
- #define _TIM1_CNTRL _SFR(uint8_t, TIM1_AddressBase+0x0F)

  TIM1 counter register low byte.
- #define _TIM1_PSCRH _SFR(uint8_t, TIM1_AddressBase+0x10)

  TIM1 clock prescaler register high byte.
- #define _TIM1_PSCRL _SFR(uint8_t, TIM1_AddressBase+0x11)

  TIM1 clock prescaler register low byte.
- #define _TIM1_ARRH _SFR(uint8_t, TIM1_AddressBase+0x12)

  TIM1 auto-reload register high byte.
- #define _TIM1_ARRL _SFR(uint8_t, TIM1_AddressBase+0x13)

  TIM1 auto-reload register low byte.
- #define _TIM1_RCR _SFR(uint8_t, TIM1_AddressBase+0x14)

TIM1 Repetition counter.

- #define _TIM1_CCR1H _SFR(uint8_t, TIM1_AddressBase+0x15)

  TIM1 16-bit capture/compare value 1 high byte.
- #define _TIM1_CCR1L _SFR(uint8_t, TIM1_AddressBase+0x16)

  TIM1 16-bit capture/compare value 1 low byte.
- #define _TIM1_CCR2H _SFR(uint8_t, TIM1_AddressBase+0x17)

  TIM1 16-bit capture/compare value 2 high byte.
- #define _TIM1_CCR2L _SFR(uint8_t, TIM1_AddressBase+0x18)

  TIM1 16-bit capture/compare value 2 low byte.
- #define _TIM1_CCR3H _SFR(uint8_t, TIM1_AddressBase+0x19)

  TIM1 16-bit capture/compare value 3 high byte.
- #define _TIM1_CCR3L _SFR(uint8_t, TIM1_AddressBase+0x1A)

  TIM1 16-bit capture/compare value 3 low byte.
- #define _TIM1_CCR4H _SFR(uint8_t, TIM1_AddressBase+0x1B)

  TIM1 16-bit capture/compare value 4 high byte.
- #define _TIM1_CCR4L _SFR(uint8_t, TIM1_AddressBase+0x1C)

  TIM1 16-bit capture/compare value 4 low byte.
- #define _TIM1_BKR _SFR(uint8_t, TIM1_AddressBase+0x1D)

  TIM1 Break register.
- #define _TIM1_DTR _SFR(uint8_t, TIM1_AddressBase+0x1E)

  TIM1 Dead-time register.
- #define _TIM1_OISR _SFR(uint8_t, TIM1_AddressBase+0x1F)

  TIM1 Output idle state register.
- #define _TIM1_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM1 control register 1 reset value.
- #define _TIM1_CR2_RESET_VALUE ((uint8_t) 0x00)

  TIM1 control register 2 reset value.
- #define _TIM1_SMCR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Slave mode control register reset value.
- #define _TIM1_ETR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 External trigger register reset value.
- #define _TIM1_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM1 interrupt enable register reset value.
- #define _TIM1_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM1 status register 1 reset value.
- #define _TIM1_SR2_RESET_VALUE ((uint8_t) 0x00)

  TIM1 status register 2 reset value.
- #define _TIM1_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Event generation register reset value.
- #define _TIM1_CCMR1_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Capture/compare mode register 1 reset value.
- #define _TIM1_CCMR2_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Capture/compare mode register 2 reset value.
- #define _TIM1_CCMR3_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Capture/compare mode register 3 reset value.
- #define _TIM1_CCMR4_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Capture/compare mode register 4 reset value.
- #define _TIM1_CCER1_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Capture/compare enable register 1 reset value.
- #define _TIM1_CCER2_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Capture/compare enable register 2 reset value.

- #define _TIM1_CNTRH_RESET_VALUE ((uint8_t) 0x00)

  TIM1 counter register high byte reset value.
- #define _TIM1_CNTRL_RESET_VALUE ((uint8_t) 0x00)

  TIM1 counter register low byte reset value.
- #define _TIM1_PSCRH_RESET_VALUE ((uint8_t) 0x00)

  TIM1 clock prescaler register high byte reset value.
- #define _TIM1_PSCRL_RESET_VALUE ((uint8_t) 0x00)

  TIM1 clock prescaler register low byte reset value.
- #define _TIM1_ARRH_RESET_VALUE ((uint8_t) 0xFF)

  TIM1 auto-reload register high byte reset value.
- #define _TIM1_ARRL_RESET_VALUE ((uint8_t) 0xFF)

  TIM1 auto-reload register low byte reset value.
- #define _TIM1_RCR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Repetition counter reset value.
- #define _TIM1_CCR1H_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 1 high byte reset value.
- #define _TIM1_CCR1L_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 1 low byte reset value.
- #define _TIM1_CCR2H_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 2 high byte reset value.
- #define _TIM1_CCR2L_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 2 low byte reset value.
- #define _TIM1_CCR3H_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 3 high byte reset value.
- #define _TIM1_CCR3L_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 3 low byte reset value.
- #define _TIM1_CCR4H_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 4 high byte reset value.
- #define _TIM1_CCR4L_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 4 low byte reset value.
- #define _TIM1_BKR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Break register reset value.
- #define _TIM1_DTR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Dead-time register reset value.
- #define _TIM1_OISR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Output idle state register reset value.
- #define _TIM1_CEN ((uint8_t) (0x01 << 0))

  TIM1 Counter enable [0] (in _TIM1_CR1)
- #define _TIM1_UDIS ((uint8_t) (0x01 << 1))

  TIM1 Update disable [0] (in _TIM1_CR1)
- #define _TIM1_URS ((uint8_t) (0x01 << 2))

  TIM1 Update request source [0] (in _TIM1_CR1)
- #define _TIM1_OPM ((uint8_t) (0x01 << 3))

  TIM1 One-pulse mode [0] (in _TIM1_CR1)
- #define _TIM1_DIR ((uint8_t) (0x01 << 4))

  TIM1 Direction [0] (in _TIM1_CR1)
- #define _TIM1_CMS ((uint8_t) (0x03 << 5))

  TIM1 Center-aligned mode selection [1:0] (in _TIM1_CR1)
- #define _TIM1_CMS0 ((uint8_t) (0x01 << 5))

  TIM1 Center-aligned mode selection [0] (in _TIM1_CR1)
- #define _TIM1_CMS1 ((uint8_t) (0x01 << 6))

TIM1 Center-aligned mode selection [1] (in _TIM1_CR1)

- #define _TIM1_ARPE ((uint8_t) (0x01 << 7))

  TIM1 Auto-reload preload enable [0] (in _TIM1_CR1)
- #define _TIM1_CCPC ((uint8_t) (0x01 << 0))

  TIM1 Capture/compare preloaded control [0] (in _TIM1_CR2)
- #define _TIM1_COMS ((uint8_t) (0x01 << 2))

  TIM1 Capture/compare control update selection [0] (in _TIM1_CR2)
- #define _TIM1_MMS ((uint8_t) (0x07 << 4))

  TIM1 Master mode selection [2:0] (in _TIM1_CR2)
- #define _TIM1_MMS0 ((uint8_t) (0x01 << 4))

  TIM1 Master mode selection [0] (in _TIM1_CR2)
- #define _TIM1_MMS1 ((uint8_t) (0x01 << 5))

  TIM1 Master mode selection [1] (in _TIM1_CR2)
- #define _TIM1_MMS2 ((uint8_t) (0x01 << 6))

  TIM1 Master mode selection [2] (in _TIM1_CR2)
- #define _TIM1_SMS ((uint8_t) (0x07 << 0))

  TIM1 Clock/trigger/slave mode selection [2:0] (in _TIM1_SMCR)
- #define _TIM1_SMS0 ((uint8_t) (0x01 << 0))

  TIM1 Clock/trigger/slave mode selection [0] (in _TIM1_SMCR)
- #define _TIM1_SMS1 ((uint8_t) (0x01 << 1))

  TIM1 Clock/trigger/slave mode selection [1] (in _TIM1_SMCR)
- #define _TIM1_SMS2 ((uint8_t) (0x01 << 2))

  TIM1 Clock/trigger/slave mode selection [2] (in _TIM1_SMCR)
- #define _TIM1_TS ((uint8_t) (0x07 << 4))

  TIM1 Trigger selection [2:0] (in _TIM1_SMCR)
- #define _TIM1_TS0 ((uint8_t) (0x01 << 4))

  TIM1 Trigger selection [0] (in _TIM1_SMCR)
- #define _TIM1_TS1 ((uint8_t) (0x01 << 5))

  TIM1 Trigger selection [1] (in _TIM1_SMCR)
- #define _TIM1_TS2 ((uint8_t) (0x01 << 6))

  TIM1 Trigger selection [2] (in _TIM1_SMCR)
- #define _TIM1_MSM ((uint8_t) (0x01 << 7))

  TIM1 Master/slave mode [0] (in _TIM1_SMCR)
- #define _TIM1_ETF ((uint8_t) (0x0F << 0))

  TIM1 External trigger filter [3:0] (in _TIM1_ETR)
- #define _TIM1_ETF0 ((uint8_t) (0x01 << 0))

  TIM1 External trigger filter [0] (in _TIM1_ETR)
- #define _TIM1_ETF1 ((uint8_t) (0x01 << 1))

  TIM1 External trigger filter [1] (in _TIM1_ETR)
- #define _TIM1_ETF2 ((uint8_t) (0x01 << 2))

  TIM1 External trigger filter [2] (in _TIM1_ETR)
- #define _TIM1_ETF3 ((uint8_t) (0x01 << 3))

  TIM1 External trigger filter [3] (in _TIM1_ETR)
- #define _TIM1_ETPS ((uint8_t) (0x03 << 4))

  TIM1 External trigger prescaler [1:0] (in _TIM1_ETR)
- #define _TIM1_ETPS0 ((uint8_t) (0x01 << 4))

  TIM1 External trigger prescaler [0] (in _TIM1_ETR)
- #define _TIM1_ETPS1 ((uint8_t) (0x01 << 5))

  TIM1 External trigger prescaler [1] (in _TIM1_ETR)
- #define _TIM1_ECE ((uint8_t) (0x01 << 6))

  TIM1 External clock enable [0] (in _TIM1_ETR)

- #define _TIM1_ETP ((uint8_t) (0x01 << 7))
  TIM1 External trigger polarity [0] (in _TIM1_ETR)
- #define _TIM1_UIE ((uint8_t) (0x01 << 0))
  TIM1 Update interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_CC1IE ((uint8_t) (0x01 << 1))
  TIM1 Capture/compare 1 interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_CC2IE ((uint8_t) (0x01 << 2))
  TIM1 Capture/compare 2 interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_CC3IE ((uint8_t) (0x01 << 3))
  TIM1 Capture/compare 3 interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_CC4IE ((uint8_t) (0x01 << 4))
  TIM1 Capture/compare 4 interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_COMIE ((uint8_t) (0x01 << 5))
  TIM1 Commutation interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_TIE ((uint8_t) (0x01 << 6))
  TIM1 Trigger interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_BIE ((uint8_t) (0x01 << 7))
  TIM1 Break interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_UIF ((uint8_t) (0x01 << 0))
  TIM1 Update interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_CC1IF ((uint8_t) (0x01 << 1))
  TIM1 Capture/compare 1 interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_CC2IF ((uint8_t) (0x01 << 2))
  TIM1 Capture/compare 2 interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_CC3IF ((uint8_t) (0x01 << 3))
  TIM1 Capture/compare 3 interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_CC4IF ((uint8_t) (0x01 << 4))
  TIM1 Capture/compare 4 interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_COMIF ((uint8_t) (0x01 << 5))
  TIM1 Commutation interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_TIF ((uint8_t) (0x01 << 6))
  TIM1 Trigger interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_BIF ((uint8_t) (0x01 << 7))
  TIM1 Break interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_CC1OF ((uint8_t) (0x01 << 1))
  TIM1 Capture/compare 1 overcapture flag [0] (in _TIM1_SR2)
- #define _TIM1_CC2OF ((uint8_t) (0x01 << 2))
  TIM1 Capture/compare 2 overcapture flag [0] (in _TIM1_SR2)
- #define _TIM1_CC3OF ((uint8_t) (0x01 << 3))
  TIM1 Capture/compare 3 overcapture flag [0] (in _TIM1_SR2)
- #define _TIM1_CC4OF ((uint8_t) (0x01 << 4))
  TIM1 Capture/compare 4 overcapture flag [0] (in _TIM1_SR2)
- #define _TIM1_UG ((uint8_t) (0x01 << 0))
  TIM1 Update generation [0] (in _TIM1_EGR)
- #define _TIM1_CC1G ((uint8_t) (0x01 << 1))
  TIM1 Capture/compare 1 generation [0] (in _TIM1_EGR)
- #define _TIM1_CC2G ((uint8_t) (0x01 << 2))
  TIM1 Capture/compare 2 generation [0] (in _TIM1_EGR)
- #define _TIM1_CC3G ((uint8_t) (0x01 << 3))
  TIM1 Capture/compare 3 generation [0] (in _TIM1_EGR)
- #define _TIM1_CC4G ((uint8_t) (0x01 << 4))

TIM1 Capture/compare 4 generation [0] (in _TIM1_EGR)

- #define _TIM1_COMG ((uint8_t) (0x01 << 5))

  TIM1 Capture/compare control update generation [0] (in _TIM1_EGR)
- #define _TIM1_TG ((uint8_t) (0x01 << 6))

  TIM1 Trigger generation [0] (in _TIM1_EGR)
- #define _TIM1_BG ((uint8_t) (0x01 << 7))

  TIM1 Break generation [0] (in _TIM1_EGR)
- #define _TIM1_CC1S ((uint8_t) (0x03 << 0))

  TIM1 Compare 1 selection [1:0] (in _TIM1_CCMR1)
- #define _TIM1_CC1S0 ((uint8_t) (0x01 << 0))

  TIM1 Compare 1 selection [0] (in _TIM1_CCMR1)
- #define _TIM1_CC1S1 ((uint8_t) (0x01 << 1))

  TIM1 Compare 1 selection [1] (in _TIM1_CCMR1)
- #define _TIM1_OC1FE ((uint8_t) (0x01 << 2))

  TIM1 Output compare 1 fast enable [0] (in _TIM1_CCMR1)
- #define _TIM1_OC1PE ((uint8_t) (0x01 << 3))

  TIM1 Output compare 1 preload enable [0] (in _TIM1_CCMR1)
- #define _TIM1_OC1M ((uint8_t) (0x07 << 4))

  TIM1 Output compare 1 mode [2:0] (in _TIM1_CCMR1)
- #define _TIM1_OC1M0 ((uint8_t) (0x01 << 4))

  TIM1 Output compare 1 mode [0] (in _TIM1_CCMR1)
- #define _TIM1_OC1M1 ((uint8_t) (0x01 << 5))

  TIM1 Output compare 1 mode [1] (in _TIM1_CCMR1)
- #define _TIM1_OC1M2 ((uint8_t) (0x01 << 6))

  TIM1 Output compare 1 mode [2] (in _TIM1_CCMR1)
- #define _TIM1_OC1CE ((uint8_t) (0x01 << 7))

  TIM1 Output compare 1 clear enable [0] (in _TIM1_CCMR1)
- #define _TIM1_IC1PSC ((uint8_t) (0x03 << 2))

  TIM1 Input capture 1 prescaler [1:0] (in _TIM1_CCMR1)
- #define _TIM1_IC1PSC0 ((uint8_t) (0x01 << 2))

  TIM1 Input capture 1 prescaler [0] (in _TIM1_CCMR1)
- #define _TIM1_IC1PSC1 ((uint8_t) (0x01 << 3))

  TIM1 Input capture 1 prescaler [1] (in _TIM1_CCMR1)
- #define _TIM1_IC1F ((uint8_t) (0x0F << 4))

  TIM1 Output compare 1 mode [3:0] (in _TIM1_CCMR1)
- #define _TIM1_IC1F0 ((uint8_t) (0x01 << 4))

  TIM1 Input capture 1 filter [0] (in _TIM1_CCMR1)
- #define _TIM1_IC1F1 ((uint8_t) (0x01 << 5))

  TIM1 Input capture 1 filter [1] (in _TIM1_CCMR1)
- #define _TIM1_IC1F2 ((uint8_t) (0x01 << 6))

  TIM1 Input capture 1 filter [2] (in _TIM1_CCMR1)
- #define _TIM1_IC1F3 ((uint8_t) (0x01 << 7))

  TIM1 Input capture 1 filter [3] (in _TIM1_CCMR1)
- #define _TIM1_CC2S ((uint8_t) (0x03 << 0))

  TIM1 Compare 2 selection [1:0] (in _TIM1_CCMR2)
- #define _TIM1_CC2S0 ((uint8_t) (0x01 << 0))

  TIM1 Compare 2 selection [0] (in _TIM1_CCMR2)
- #define _TIM1_CC2S1 ((uint8_t) (0x01 << 1))

  TIM1 Compare 2 selection [1] (in _TIM1_CCMR2)
- #define _TIM1_OC2FE ((uint8_t) (0x01 << 2))

  TIM1 Output compare 2 fast enable [0] (in _TIM1_CCMR2)

- #define _TIM1_OC2PE ((uint8_t) (0x01 << 3))
  TIM1 Output compare 2 preload enable [0] (in _TIM1_CCMR2)
- #define _TIM1_OC2M ((uint8_t) (0x07 << 4))
  TIM1 Output compare 2 mode [2:0] (in _TIM1_CCMR2)
- #define _TIM1_OC2M0 ((uint8_t) (0x01 << 4))
  TIM1 Output compare 2 mode [0] (in _TIM1_CCMR2)
- #define _TIM1_OC2M1 ((uint8_t) (0x01 << 5))
  TIM1 Output compare 2 mode [1] (in _TIM1_CCMR2)
- #define _TIM1_OC2M2 ((uint8_t) (0x01 << 6))
  TIM1 Output compare 2 mode [2] (in _TIM1_CCMR2)
- #define _TIM1_OC2CE ((uint8_t) (0x01 << 7))
  TIM1 Output compare 2 clear enable [0] (in _TIM1_CCMR2)
- #define _TIM1_IC2PSC ((uint8_t) (0x03 << 2))
  TIM1 Input capture 2 prescaler [1:0] (in _TIM1_CCMR2)
- #define _TIM1_IC2PSC0 ((uint8_t) (0x01 << 2))
  TIM1 Input capture 2 prescaler [0] (in _TIM1_CCMR2)
- #define _TIM1_IC2PSC1 ((uint8_t) (0x01 << 3))
  TIM1 Input capture 2 prescaler [1] (in _TIM1_CCMR2)
- #define _TIM1_IC2F ((uint8_t) (0x0F << 4))
  TIM1 Output compare 2 mode [3:0] (in _TIM1_CCMR2)
- #define _TIM1_IC2F0 ((uint8_t) (0x01 << 4))
  TIM1 Input capture 2 filter [0] (in _TIM1_CCMR2)
- #define _TIM1_IC2F1 ((uint8_t) (0x01 << 5))
  TIM1 Input capture 2 filter [1] (in _TIM1_CCMR2)
- #define _TIM1_IC2F2 ((uint8_t) (0x01 << 6))
  TIM1 Input capture 2 filter [2] (in _TIM1_CCMR2)
- #define _TIM1_IC2F3 ((uint8_t) (0x01 << 7))
  TIM1 Input capture 2 filter [3] (in _TIM1_CCMR2)
- #define _TIM1_CC3S ((uint8_t) (0x03 << 0))
  TIM1 Compare 3 selection [1:0] (in _TIM1_CCMR3)
- #define _TIM1_CC3S0 ((uint8_t) (0x01 << 0))
  TIM1 Compare 3 selection [0] (in _TIM1_CCMR3)
- #define _TIM1_CC3S1 ((uint8_t) (0x01 << 1))
  TIM1 Compare 3 selection [1] (in _TIM1_CCMR3)
- #define _TIM1_OC3FE ((uint8_t) (0x01 << 2))
  TIM1 Output compare 3 fast enable [0] (in _TIM1_CCMR3)
- #define _TIM1_OC3PE ((uint8_t) (0x01 << 3))
  TIM1 Output compare 3 preload enable [0] (in _TIM1_CCMR3)
- #define _TIM1_OC3M ((uint8_t) (0x07 << 4))
  TIM1 Output compare 3 mode [2:0] (in _TIM1_CCMR3)
- #define _TIM1_OC3M0 ((uint8_t) (0x01 << 4))
  TIM1 Output compare 3 mode [0] (in _TIM1_CCMR3)
- #define _TIM1_OC3M1 ((uint8_t) (0x01 << 5))
  TIM1 Output compare 3 mode [1] (in _TIM1_CCMR3)
- #define _TIM1_OC3M2 ((uint8_t) (0x01 << 6))
  TIM1 Output compare 3 mode [2] (in _TIM1_CCMR3)
- #define _TIM1_OC3CE ((uint8_t) (0x01 << 7))
  TIM1 Output compare 3 clear enable [0] (in _TIM1_CCMR3)
- #define _TIM1_IC3PSC ((uint8_t) (0x03 << 2))
  TIM1 Input capture 3 prescaler [1:0] (in _TIM1_CCMR3)
- #define _TIM1_IC3PSC0 ((uint8_t) (0x01 << 2))

TIM1 Input capture 3 prescaler [0] (in _TIM1_CCMR3)

- #define _TIM1_IC3PSC1 ((uint8_t) (0x01 << 3))

  TIM1 Input capture 3 prescaler [1] (in _TIM1_CCMR3)
- #define _TIM1_IC3F ((uint8_t) (0x0F << 4))

  TIM1 Output compare 3 mode [3:0] (in _TIM1_CCMR3)
- #define _TIM1_IC3F0 ((uint8_t) (0x01 << 4))

  TIM1 Input capture 3 filter [0] (in _TIM1_CCMR3)
- #define _TIM1_IC3F1 ((uint8_t) (0x01 << 5))

  TIM1 Input capture 3 filter [1] (in _TIM1_CCMR3)
- #define _TIM1_IC3F2 ((uint8_t) (0x01 << 6))

  TIM1 Input capture 3 filter [2] (in _TIM1_CCMR3)
- #define _TIM1_IC3F3 ((uint8_t) (0x01 << 7))

  TIM1 Input capture 3 filter [3] (in _TIM1_CCMR3)
- #define _TIM1_CC4S ((uint8_t) (0x03 << 0))

  TIM1 Compare 4 selection [1:0] (in _TIM1_CCMR4)
- #define _TIM1_CC4S0 ((uint8_t) (0x01 << 0))

  TIM1 Compare 4 selection [0] (in _TIM1_CCMR4)
- #define _TIM1_CC4S1 ((uint8_t) (0x01 << 1))

  TIM1 Compare 4 selection [1] (in _TIM1_CCMR4)
- #define _TIM1_OC4FE ((uint8_t) (0x01 << 2))

  TIM1 Output compare 4 fast enable [0] (in _TIM1_CCMR4)
- #define _TIM1_OC4PE ((uint8_t) (0x01 << 3))

  TIM1 Output compare 4 preload enable [0] (in _TIM1_CCMR4)
- #define _TIM1_OC4M ((uint8_t) (0x07 << 4))

  TIM1 Output compare 4 mode [2:0] (in _TIM1_CCMR4)
- #define _TIM1_OC4M0 ((uint8_t) (0x01 << 4))

  TIM1 Output compare 4 mode [0] (in _TIM1_CCMR4)
- #define _TIM1_OC4M1 ((uint8_t) (0x01 << 5))

  TIM1 Output compare 4 mode [1] (in _TIM1_CCMR4)
- #define _TIM1_OC4M2 ((uint8_t) (0x01 << 6))

  TIM1 Output compare 4 mode [2] (in _TIM1_CCMR4)
- #define _TIM1_OC4CE ((uint8_t) (0x01 << 7))

  TIM1 Output compare 4 clear enable [0] (in _TIM1_CCMR4)
- #define _TIM1_IC4PSC ((uint8_t) (0x03 << 2))

  TIM1 Input capture 4 prescaler [1:0] (in _TIM1_CCMR4)
- #define _TIM1_IC4PSC0 ((uint8_t) (0x01 << 2))

  TIM1 Input capture 4 prescaler [0] (in _TIM1_CCMR4)
- #define _TIM1_IC4PSC1 ((uint8_t) (0x01 << 3))

  TIM1 Input capture 4 prescaler [1] (in _TIM1_CCMR4)
- #define _TIM1_IC4F ((uint8_t) (0x0F << 4))

  TIM1 Output compare 4 mode [3:0] (in _TIM1_CCMR4)
- #define _TIM1_IC4F0 ((uint8_t) (0x01 << 4))

  TIM1 Input capture 4 filter [0] (in _TIM1_CCMR4)
- #define _TIM1_IC4F1 ((uint8_t) (0x01 << 5))

  TIM1 Input capture 4 filter [1] (in _TIM1_CCMR4)
- #define _TIM1_IC4F2 ((uint8_t) (0x01 << 6))

  TIM1 Input capture 4 filter [2] (in _TIM1_CCMR4)
- #define _TIM1_IC4F3 ((uint8_t) (0x01 << 7))

  TIM1 Input capture 4 filter [3] (in _TIM1_CCMR4)
- #define _TIM1_CC1E ((uint8_t) (0x01 << 0))

  TIM1 Capture/compare 1 output enable [0] (in _TIM1_CCER1)

- #define _TIM1_CC1P ((uint8_t) (0x01 << 1))

  TIM1 Capture/compare 1 output polarity [0] (in _TIM1_CCER1)
- #define _TIM1_CC1NE ((uint8_t) (0x01 << 2))

  TIM1 Capture/compare 1 complementary output enable [0] (in _TIM1_CCER1)
- #define _TIM1_CC1NP ((uint8_t) (0x01 << 3))

  TIM1 Capture/compare 1 complementary output polarity [0] (in _TIM1_CCER1)
- #define _TIM1_CC2E ((uint8_t) (0x01 << 4))

  TIM1 Capture/compare 2 output enable [0] (in _TIM1_CCER1)
- #define _TIM1_CC2P ((uint8_t) (0x01 << 5))

  TIM1 Capture/compare 2 output polarity [0] (in _TIM1_CCER1)
- #define _TIM1_CC2NE ((uint8_t) (0x01 << 6))

  TIM1 Capture/compare 2 complementary output enable [0] (in _TIM1_CCER1)
- #define _TIM1_CC2NP ((uint8_t) (0x01 << 7))

  TIM1 Capture/compare 2 complementary output polarity [0] (in _TIM1_CCER1)
- #define _TIM1_CC3E ((uint8_t) (0x01 << 0))

  TIM1 Capture/compare 3 output enable [0] (in _TIM1_CCER2)
- #define _TIM1_CC3P ((uint8_t) (0x01 << 1))

  TIM1 Capture/compare 3 output polarity [0] (in _TIM1_CCER2)
- #define _TIM1_CC3NE ((uint8_t) (0x01 << 2))

  TIM1 Capture/compare 3 complementary output enable [0] (in _TIM1_CCER2)
- #define _TIM1_CC3NP ((uint8_t) (0x01 << 3))

  TIM1 Capture/compare 3 complementary output polarity [0] (in _TIM1_CCER2)
- #define _TIM1_CC4E ((uint8_t) (0x01 << 4))

  TIM1 Capture/compare 4 output enable [0] (in _TIM1_CCER2)
- #define _TIM1_CC4P ((uint8_t) (0x01 << 5))

  TIM1 Capture/compare 4 output polarity [0] (in _TIM1_CCER2)
- #define _TIM1_LOCK ((int8_t) (0x03 << 0))

  TIM1 Lock configuration [1:0] (in _TIM1_BKR)
- #define _TIM1_LOCK0 ((uint8_t) (0x01 << 0))

  TIM1 Lock configuration [0] (in _TIM1_BKR)
- #define _TIM1_LOCK1 ((uint8_t) (0x01 << 1))

  TIM1 Lock configuration [1] (in _TIM1_BKR)
- #define _TIM1_OSSI ((uint8_t) (0x01 << 2))

  TIM1 Off state selection for idle mode [0] (in _TIM1_BKR)
- #define _TIM1_OSSR ((uint8_t) (0x01 << 3))

  TIM1 Off state selection for Run mode [0] (in _TIM1_BKR)
- #define _TIM1_BKE ((uint8_t) (0x01 << 4))

  TIM1 Break enable [0] (in _TIM1_BKR)
- #define _TIM1_BKP ((uint8_t) (0x01 << 5))

  TIM1 Break polarity [0] (in _TIM1_BKR)
- #define _TIM1_AOE ((uint8_t) (0x01 << 6))

  TIM1 Automatic output enable [0] (in _TIM1_BKR)
- #define _TIM1_MOE ((uint8_t) (0x01 << 7))

  TIM1 Main output enable [0] (in _TIM1_BKR)
- #define _TIM1_OIS1 ((uint8_t) (0x01 << 0))

  TIM1 Output idle state 1 (OC1 output) [0] (in _TIM1_OISR)
- #define _TIM1_OIS1N ((uint8_t) (0x01 << 1))

  TIM1 Output idle state 1 (OC1N output) [0] (in _TIM1_OISR)
- #define _TIM1_OIS2 ((uint8_t) (0x01 << 2))

  TIM1 Output idle state 2 (OC2 output) [0] (in _TIM1_OISR)
- #define _TIM1_OIS2N ((uint8_t) (0x01 << 3))

TIM1 Output idle state 2 (OC2N output) [0] (in _TIM1_OISR)

- #define _TIM1_OIS3 ((uint8_t) (0x01 << 4))

  TIM1 Output idle state 3 (OC3 output) [0] (in _TIM1_OISR)
- #define _TIM1_OIS3N ((uint8_t) (0x01 << 5))

  TIM1 Output idle state 3 (OC3N output) [0] (in _TIM1_OISR)
- #define _TIM1_OIS4 ((uint8_t) (0x01 << 6))

  TIM1 Output idle state 4 (OC4 output) [0] (in _TIM1_OISR)
- #define _TIM2 _SFR(TIM2_t, TIM2_AddressBase)

  TIM2 struct/bit access.
- #define _TIM2_CR1 _SFR(uint8_t, TIM2_AddressBase+0x00)

  TIM2 control register 1.
- #define _TIM2_IER _SFR(uint8_t, TIM2_AddressBase+0x01)

  TIM2 interrupt enable register.
- #define _TIM2_SR1 _SFR(uint8_t, TIM2_AddressBase+0x02)

  TIM2 status register 1.
- #define _TIM2_SR2 _SFR(uint8_t, TIM2_AddressBase+0x03)

  TIM2 status register 2.
- #define _TIM2_EGR _SFR(uint8_t, TIM2_AddressBase+0x04)

  TIM2 Event generation register.
- #define _TIM2_CCMR1 _SFR(uint8_t, TIM2_AddressBase+0x05)

  TIM2 Capture/compare mode register 1.
- #define _TIM2_CCMR2 _SFR(uint8_t, TIM2_AddressBase+0x06)

  TIM2 Capture/compare mode register 2.
- #define _TIM2_CCMR3 _SFR(uint8_t, TIM2_AddressBase+0x07)

  TIM2 Capture/compare mode register 3.
- #define _TIM2_CCER1 _SFR(uint8_t, TIM2_AddressBase+0x08)

  TIM2 Capture/compare enable register 1.
- #define _TIM2_CCER2 _SFR(uint8_t, TIM2_AddressBase+0x09)

  TIM2 Capture/compare enable register 2.
- #define _TIM2_CNTRH _SFR(uint8_t, TIM2_AddressBase+0x0A)

  TIM2 counter register high byte.
- #define _TIM2_CNTRL _SFR(uint8_t, TIM2_AddressBase+0x0B)

  TIM2 counter register low byte.
- #define _TIM2_PSCR _SFR(uint8_t, TIM2_AddressBase+0x0C)

  TIM2 clock prescaler register.
- #define _TIM2_ARRH _SFR(uint8_t, TIM2_AddressBase+0x0D)

  TIM2 auto-reload register high byte.
- #define _TIM2_ARRL _SFR(uint8_t, TIM2_AddressBase+0x0E)

  TIM2 auto-reload register low byte.
- #define _TIM2_CCR1H _SFR(uint8_t, TIM2_AddressBase+0x0F)

  TIM2 16-bit capture/compare value 1 high byte.
- #define _TIM2_CCR1L _SFR(uint8_t, TIM2_AddressBase+0x10)

  TIM2 16-bit capture/compare value 1 low byte.
- #define _TIM2_CCR2H _SFR(uint8_t, TIM2_AddressBase+0x11)

  TIM2 16-bit capture/compare value 2 high byte.
- #define _TIM2_CCR2L _SFR(uint8_t, TIM2_AddressBase+0x12)

  TIM2 16-bit capture/compare value 2 low byte.
- #define _TIM2_CCR3H _SFR(uint8_t, TIM2_AddressBase+0x13)

  TIM2 16-bit capture/compare value 3 high byte.
- #define _TIM2_CCR3L _SFR(uint8_t, TIM2_AddressBase+0x14)

  TIM2 16-bit capture/compare value 3 low byte.

- #define _TIM2_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 control register 1 reset value.
- #define _TIM2_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM2 interrupt enable register reset value.
- #define _TIM2_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 status register 1 reset value.
- #define _TIM2_SR2_RESET_VALUE ((uint8_t) 0x00)

  TIM2 status register 2 reset value.
- #define _TIM2_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Event generation register reset value.
- #define _TIM2_CCMR1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare mode register 1 reset value.
- #define _TIM2_CCMR2_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare mode register 2 reset value.
- #define _TIM2_CCMR3_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare mode register 3 reset value.
- #define _TIM2_CCER1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare enable register 1 reset value.
- #define _TIM2_CCER2_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare enable register 2 reset value.
- #define _TIM2_CNTRH_RESET_VALUE ((uint8_t) 0x00)

  TIM2 counter register high byte reset value.
- #define _TIM2_CNTRL_RESET_VALUE ((uint8_t) 0x00)

  TIM2 counter register low byte reset value.
- #define _TIM2_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 clock prescaler register reset value.
- #define _TIM2_ARRH_RESET_VALUE ((uint8_t) 0xFF)

  TIM2 auto-reload register high byte reset value.
- #define _TIM2_ARRL_RESET_VALUE ((uint8_t) 0xFF)

  TIM2 auto-reload register low byte reset value.
- #define _TIM2_CCR1H_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 1 high byte reset value.
- #define _TIM2_CCR1L_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 1 low byte reset value.
- #define _TIM2_CCR2H_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 2 high byte reset value.
- #define _TIM2_CCR2L_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 2 low byte reset value.
- #define _TIM2_CCR3H_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 3 high byte reset value.
- #define _TIM2_CCR3L_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 3 low byte reset value.
- #define _TIM2_CEN ((uint8_t) (0x01 << 0))

  TIM2 Counter enable [0] (in _TIM2_CR1)
- #define _TIM2_UDIS ((uint8_t) (0x01 << 1))

  TIM2 Update disable [0] (in _TIM2_CR1)
- #define _TIM2_URS ((uint8_t) (0x01 << 2))

  TIM2 Update request source [0] (in _TIM2_CR1)
- #define _TIM2_OPM ((uint8_t) (0x01 << 3))

  TIM2 One-pulse mode [0] (in _TIM2_CR1)
- #define _TIM2_ARPE ((uint8_t) (0x01 << 7))

TIM2 Auto-reload preload enable [0] (in _TIM2_CR1)

- #define _TIM2_UIE ((uint8_t) (0x01 << 0))

  TIM2 Update interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_CC1IE ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_CC2IE ((uint8_t) (0x01 << 2))

  TIM2 Capture/compare 2 interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_CC3IE ((uint8_t) (0x01 << 3))

  TIM2 Capture/compare 3 interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_UIF ((uint8_t) (0x01 << 0))

  TIM2 Update interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC1IF ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC2IF ((uint8_t) (0x01 << 2))

  TIM2 Capture/compare 2 interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC3IF ((uint8_t) (0x01 << 3))

  TIM2 Capture/compare 3 interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC1OF ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 overcapture flag [0] (in _TIM2_SR2)
- #define _TIM2_CC2OF ((uint8_t) (0x01 << 2))

  TIM2 Capture/compare 2 overcapture flag [0] (in _TIM2_SR2)
- #define _TIM2_CC3OF ((uint8_t) (0x01 << 3))

  TIM2 Capture/compare 3 overcapture flag [0] (in _TIM2_SR2)
- #define _TIM2_UG ((uint8_t) (0x01 << 0))

  TIM2 Update generation [0] (in _TIM2_EGR)
- #define _TIM2_CC1G ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 generation [0] (in _TIM2_EGR)
- #define _TIM2_CC2G ((uint8_t) (0x01 << 2))

  TIM2 Capture/compare 2 generation [0] (in _TIM2_EGR)
- #define _TIM2_CC3G ((uint8_t) (0x01 << 3))

  TIM2 Capture/compare 3 generation [0] (in _TIM2_EGR)
- #define _TIM2_CC1S ((uint8_t) (0x03 << 0))

  TIM2 Compare 1 selection [1:0] (in _TIM2_CCMR1)
- #define _TIM2_CC1S0 ((uint8_t) (0x01 << 0))

  TIM2 Compare 1 selection [0] (in _TIM2_CCMR1)
- #define _TIM2_CC1S1 ((uint8_t) (0x01 << 1))

  TIM2 Compare 1 selection [1] (in _TIM2_CCMR1)
- #define _TIM2_OC1PE ((uint8_t) (0x01 << 3))

  TIM2 Output compare 1 preload enable [0] (in _TIM2_CCMR1)
- #define _TIM2_OC1M ((uint8_t) (0x07 << 4))

  TIM2 Output compare 1 mode [2:0] (in _TIM2_CCMR1)
- #define _TIM2_OC1M0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 1 mode [0] (in _TIM2_CCMR1)
- #define _TIM2_OC1M1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 1 mode [1] (in _TIM2_CCMR1)
- #define _TIM2_OC1M2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 1 mode [2] (in _TIM2_CCMR1)
- #define _TIM2_IC1PSC ((uint8_t) (0x03 << 2))

  TIM2 Input capture 1 prescaler [1:0] (in _TIM2_CCMR1)
- #define _TIM2_IC1PSC0 ((uint8_t) (0x01 << 2))

  TIM2 Input capture 1 prescaler [0] (in _TIM2_CCMR1)

- #define _TIM2_IC1PSC1 ((uint8_t) (0x01 << 3))

  TIM2 Input capture 1 prescaler [1] (in _TIM2_CCMR1)
- #define _TIM2_IC1F ((uint8_t) (0x0F << 4))

  TIM2 Output compare 1 mode [3:0] (in _TIM2_CCMR1)
- #define _TIM2_IC1F0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 1 mode [0] (in _TIM2_CCMR1)
- #define _TIM2_IC1F1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 1 mode [1] (in _TIM2_CCMR1)
- #define _TIM2_IC1F2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 1 mode [2] (in _TIM2_CCMR1)
- #define _TIM2_IC1F3 ((uint8_t) (0x01 << 7))

  TIM2 Output compare 1 mode [3] (in _TIM2_CCMR1)
- #define _TIM2_CC2S ((uint8_t) (0x03 << 0))

  TIM2 Compare 2 selection [1:0] (in _TIM2_CCMR2)
- #define _TIM2_CC2S0 ((uint8_t) (0x01 << 0))

  TIM2 Compare 2 selection [0] (in _TIM2_CCMR2)
- #define _TIM2_CC2S1 ((uint8_t) (0x01 << 1))

  TIM2 Compare 2 selection [1] (in _TIM2_CCMR2)
- #define _TIM2_OC2PE ((uint8_t) (0x01 << 3))

  TIM2 Output compare 2 preload enable [0] (in _TIM2_CCMR2)
- #define _TIM2_OC2M ((uint8_t) (0x07 << 4))

  TIM2 Output compare 2 mode [2:0] (in _TIM2_CCMR2)
- #define _TIM2_OC2M0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 2 mode [0] (in _TIM2_CCMR2)
- #define _TIM2_OC2M1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 2 mode [1] (in _TIM2_CCMR2)
- #define _TIM2_OC2M2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 2 mode [2] (in _TIM2_CCMR2)
- #define _TIM2_IC2PSC ((uint8_t) (0x03 << 2))

  TIM2 Input capture 2 prescaler [1:0] (in _TIM2_CCMR2)
- #define _TIM2_IC2PSC0 ((uint8_t) (0x01 << 2))

  TIM2 Input capture 2 prescaler [0] (in _TIM2_CCMR2)
- #define _TIM2_IC2PSC1 ((uint8_t) (0x01 << 3))

  TIM2 Input capture 2 prescaler [1] (in _TIM2_CCMR2)
- #define _TIM2_IC2F ((uint8_t) (0x0F << 4))

  TIM2 Output compare 2 mode [3:0] (in _TIM2_CCMR2)
- #define _TIM2_IC2F0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 2 mode [0] (in _TIM2_CCMR2)
- #define _TIM2_IC2F1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 2 mode [1] (in _TIM2_CCMR2)
- #define _TIM2_IC2F2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 2 mode [2] (in _TIM2_CCMR2)
- #define _TIM2_IC2F3 ((uint8_t) (0x01 << 7))

  TIM2 Output compare 2 mode [3] (in _TIM2_CCMR2)
- #define _TIM2_CC3S ((uint8_t) (0x03 << 0))

  TIM2 Compare 3 selection [1:0] (in _TIM2_CCMR3)
- #define _TIM2_CC3S0 ((uint8_t) (0x01 << 0))

  TIM2 Compare 3 selection [0] (in _TIM2_CCMR3)
- #define _TIM2_CC3S1 ((uint8_t) (0x01 << 1))

  TIM2 Compare 3 selection [1] (in _TIM2_CCMR3)
- #define _TIM2_OC3PE ((uint8_t) (0x01 << 3))

TIM2 Output compare 3 preload enable [0] (in _TIM2_CCMR3)

- #define _TIM2_OC3M ((uint8_t) (0x07 << 4))

  TIM2 Output compare 3 mode [2:0] (in _TIM2_CCMR3)
- #define _TIM2_OC3M0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 3 mode [0] (in _TIM2_CCMR3)
- #define _TIM2_OC3M1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 3 mode [1] (in _TIM2_CCMR3)
- #define _TIM2_OC3M2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 3 mode [2] (in _TIM2_CCMR3)
- #define _TIM2_IC3PSC ((uint8_t) (0x03 << 2))

  TIM2 Input capture 3 prescaler [1:0] (in _TIM2_CCMR3)
- #define _TIM2_IC3PSC0 ((uint8_t) (0x01 << 2))

  TIM2 Input capture 3 prescaler [0] (in _TIM2_CCMR3)
- #define _TIM2_IC3PSC1 ((uint8_t) (0x01 << 3))

  TIM2 Input capture 3 prescaler [1] (in _TIM2_CCMR3)
- #define _TIM2_IC3F ((uint8_t) (0x0F << 4))

  TIM2 Output compare 3 mode [3:0] (in _TIM2_CCMR3)
- #define _TIM2_IC3F0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 3 mode [0] (in _TIM2_CCMR3)
- #define _TIM2_IC3F1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 3 mode [1] (in _TIM2_CCMR3)
- #define _TIM2_IC3F2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 3 mode [2] (in _TIM2_CCMR3)
- #define _TIM2_IC3F3 ((uint8_t) (0x01 << 7))

  TIM2 Output compare 3 mode [3] (in _TIM2_CCMR3)
- #define _TIM2_CC1E ((uint8_t) (0x01 << 0))

  TIM2 Capture/compare 1 output enable [0] (in _TIM2_CCER1)
- #define _TIM2_CC1P ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 output polarity [0] (in _TIM2_CCER1)
- #define _TIM2_CC2E ((uint8_t) (0x01 << 4))

  TIM2 Capture/compare 2 output enable [0] (in _TIM2_CCER1)
- #define _TIM2_CC2P ((uint8_t) (0x01 << 5))

  TIM2 Capture/compare 2 output polarity [0] (in _TIM2_CCER1)
- #define _TIM2_CC3E ((uint8_t) (0x01 << 0))

  TIM2 Capture/compare 3 output enable [0] (in _TIM2_CCER2)
- #define _TIM2_CC3P ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 3 output polarity [0] (in _TIM2_CCER2)
- #define _TIM2_PSC ((uint8_t) (0x0F << 0))

  TIM2 prescaler [3:0] (in _TIM2_PSCR)
- #define _TIM2_PSC0 ((uint8_t) (0x01 << 0))

  TIM2 prescaler [0] (in _TIM2_PSCR)
- #define _TIM2_PSC1 ((uint8_t) (0x01 << 1))

  TIM2 prescaler [1] (in _TIM2_PSCR)
- #define _TIM2_PSC2 ((uint8_t) (0x01 << 2))

  TIM2 prescaler [2] (in _TIM2_PSCR)
- #define _TIM2_PSC3 ((uint8_t) (0x01 << 3))

  TIM2 prescaler [3] (in _TIM2_PSCR)
- #define _TIM3 _SFR(TIM3_t, TIM3_AddressBase)

  TIM3 struct/bit access.
- #define _TIM3_CR1 _SFR(uint8_t, TIM3_AddressBase+0x00)

  TIM3 control register 1.

- #define _TIM3_IER _SFR(uint8_t, TIM3_AddressBase+0x01)

  TIM3 interrupt enable register.
- #define _TIM3_SR1 _SFR(uint8_t, TIM3_AddressBase+0x02)

  TIM3 status register 1.
- #define _TIM3_SR2 _SFR(uint8_t, TIM3_AddressBase+0x03)

  TIM3 status register 2.
- #define _TIM3_EGR _SFR(uint8_t, TIM3_AddressBase+0x04)

  TIM3 Event generation register.
- #define _TIM3_CCMR1 _SFR(uint8_t, TIM3_AddressBase+0x05)

  TIM3 Capture/compare mode register 1.
- #define _TIM3_CCMR2 _SFR(uint8_t, TIM3_AddressBase+0x06)

  TIM3 Capture/compare mode register 2.
- #define _TIM3_CCER1 _SFR(uint8_t, TIM3_AddressBase+0x08)

  TIM3 Capture/compare enable register 1.
- #define _TIM3_CNTRH _SFR(uint8_t, TIM3_AddressBase+0x0A)

  TIM3 counter register high byte.
- #define _TIM3_CNTRL _SFR(uint8_t, TIM3_AddressBase+0x0B)

  TIM3 counter register low byte.
- #define _TIM3_PSCR _SFR(uint8_t, TIM3_AddressBase+0x0C)

  TIM3 clock prescaler register.
- #define _TIM3_ARRH _SFR(uint8_t, TIM3_AddressBase+0x0D)

  TIM3 auto-reload register high byte.
- #define _TIM3_ARRL _SFR(uint8_t, TIM3_AddressBase+0x0E)

  TIM3 auto-reload register low byte.
- #define _TIM3_CCR1H _SFR(uint8_t, TIM3_AddressBase+0x0F)

  TIM3 16-bit capture/compare value 1 high byte.
- #define _TIM3_CCR1L _SFR(uint8_t, TIM3_AddressBase+0x10)

  TIM3 16-bit capture/compare value 1 low byte.
- #define _TIM3_CCR2H _SFR(uint8_t, TIM3_AddressBase+0x11)

  TIM3 16-bit capture/compare value 2 high byte.
- #define _TIM3_CCR2L _SFR(uint8_t, TIM3_AddressBase+0x12)

  TIM3 16-bit capture/compare value 2 low byte.
- #define _TIM3_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 control register 1 reset value.
- #define _TIM3_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM3 interrupt enable register reset value.
- #define _TIM3_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 status register 1 reset value.
- #define _TIM3_SR2_RESET_VALUE ((uint8_t) 0x00)

  TIM3 status register 2 reset value.
- #define _TIM3_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Event generation register reset value.
- #define _TIM3_CCMR1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Capture/compare mode register 1 reset value.
- #define _TIM3_CCMR2_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Capture/compare mode register 2 reset value.
- #define _TIM3_CCER1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Capture/compare enable register 1 reset value.
- #define _TIM3_CNTRH_RESET_VALUE ((uint8_t) 0x00)

  TIM3 counter register high byte reset value.
- #define _TIM3_CNTRL_RESET_VALUE ((uint8_t) 0x00)

TIM3 counter register low byte reset value.

- #define _TIM3_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 clock prescaler register reset value.
- #define _TIM3_ARRH_RESET_VALUE ((uint8_t) 0xFF)

  TIM3 auto-reload register high byte reset value.
- #define _TIM3_ARRL_RESET_VALUE ((uint8_t) 0xFF)

  TIM3 auto-reload register low byte reset value.
- #define _TIM3_CCR1H_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 1 high byte reset value.
- #define _TIM3_CCR1L_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 1 low byte reset value.
- #define _TIM3_CCR2H_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 2 high byte reset value.
- #define _TIM3_CCR2L_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 2 low byte reset value.
- #define _TIM3_CEN ((uint8_t) (0x01 << 0))

  TIM3 Counter enable [0] (in _TIM3_CR1)
- #define _TIM3_UDIS ((uint8_t) (0x01 << 1))

  TIM3 Update disable [0] (in _TIM3_CR1)
- #define _TIM3_URS ((uint8_t) (0x01 << 2))

  TIM3 Update request source [0] (in _TIM3_CR1)
- #define _TIM3_OPM ((uint8_t) (0x01 << 3))

  TIM3 One-pulse mode [0] (in _TIM3_CR1)
- #define _TIM3_ARPE ((uint8_t) (0x01 << 7))

  TIM3 Auto-reload preload enable [0] (in _TIM3_CR1)
- #define _TIM3_UIE ((uint8_t) (0x01 << 0))

  TIM3 Update interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_CC1IE ((uint8_t) (0x01 << 1))

  TIM3 Capture/compare 1 interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_CC2IE ((uint8_t) (0x01 << 2))

  TIM3 Capture/compare 2 interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_UIF ((uint8_t) (0x01 << 0))

  TIM3 Update interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_CC1IF ((uint8_t) (0x01 << 1))

  TIM3 Capture/compare 1 interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_CC2IF ((uint8_t) (0x01 << 2))

  TIM3 Capture/compare 2 interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_CC1OF ((uint8_t) (0x01 << 1))

  TIM3 Capture/compare 1 overcapture flag [0] (in _TIM3_SR2)
- #define _TIM3_CC2OF ((uint8_t) (0x01 << 2))

  TIM3 Capture/compare 2 overcapture flag [0] (in _TIM3_SR2)
- #define _TIM3_UG ((uint8_t) (0x01 << 0))

  TIM3 Update generation [0] (in _TIM3_EGR)
- #define _TIM3_CC1G ((uint8_t) (0x01 << 1))

  TIM3 Capture/compare 1 generation [0] (in _TIM3_EGR)
- #define _TIM3_CC2G ((uint8_t) (0x01 << 2))

  TIM3 Capture/compare 2 generation [0] (in _TIM3_EGR)
- #define _TIM3_CC1S ((uint8_t) (0x03 << 0))

  TIM3 Compare 1 selection [1:0] (in _TIM3_CCMR1)
- #define _TIM3_CC1S0 ((uint8_t) (0x01 << 0))

  TIM3 Compare 1 selection [0] (in _TIM3_CCMR1)

- #define _TIM3_CC1S1 ((uint8_t) (0x01 << 1))
  TIM3 Compare 1 selection [1] (in _TIM3_CCMR1)
- #define _TIM3_OC1PE ((uint8_t) (0x01 << 3))
  TIM3 Output compare 1 preload enable [0] (in _TIM3_CCMR1)
- #define _TIM3_OC1M ((uint8_t) (0x07 << 4))
  TIM3 Output compare 1 mode [2:0] (in _TIM3_CCMR1)
- #define _TIM3_OC1M0 ((uint8_t) (0x01 << 4))
  TIM3 Output compare 1 mode [0] (in _TIM3_CCMR1)
- #define _TIM3_OC1M1 ((uint8_t) (0x01 << 5))
  TIM3 Output compare 1 mode [1] (in _TIM3_CCMR1)
- #define _TIM3_OC1M2 ((uint8_t) (0x01 << 6))
  TIM3 Output compare 1 mode [2] (in _TIM3_CCMR1)
- #define _TIM3_IC1PSC ((uint8_t) (0x03 << 2))
  TIM3 Input capture 1 prescaler [1:0] (in _TIM3_CCMR1)
- #define _TIM3_IC1PSC0 ((uint8_t) (0x01 << 2))
  TIM3 Input capture 1 prescaler [0] (in _TIM3_CCMR1)
- #define _TIM3_IC1PSC1 ((uint8_t) (0x01 << 3))
  TIM3 Input capture 1 prescaler [1] (in _TIM3_CCMR1)
- #define _TIM3_IC1F ((uint8_t) (0x0F << 4))
  TIM3 Output compare 1 mode [3:0] (in _TIM3_CCMR1)
- #define _TIM3_IC1F0 ((uint8_t) (0x01 << 4))
  TIM3 Output compare 1 mode [0] (in _TIM3_CCMR1)
- #define _TIM3_IC1F1 ((uint8_t) (0x01 << 5))
  TIM3 Output compare 1 mode [1] (in _TIM3_CCMR1)
- #define _TIM3_IC1F2 ((uint8_t) (0x01 << 6))
  TIM3 Output compare 1 mode [2] (in _TIM3_CCMR1)
- #define _TIM3_IC1F3 ((uint8_t) (0x01 << 7))
  TIM3 Output compare 1 mode [3] (in _TIM3_CCMR1)
- #define _TIM3_CC2S ((uint8_t) (0x03 << 0))
  TIM3 Compare 2 selection [1:0] (in _TIM3_CCMR2)
- #define _TIM3_CC2S0 ((uint8_t) (0x01 << 0))
  TIM3 Compare 2 selection [0] (in _TIM3_CCMR2)
- #define _TIM3_CC2S1 ((uint8_t) (0x01 << 1))
  TIM3 Compare 2 selection [1] (in _TIM3_CCMR2)
- #define _TIM3_OC2PE ((uint8_t) (0x01 << 3))
  TIM3 Output compare 2 preload enable [0] (in _TIM3_CCMR2)
- #define _TIM3_OC2M ((uint8_t) (0x07 << 4))
  TIM3 Output compare 2 mode [2:0] (in _TIM3_CCMR2)
- #define _TIM3_OC2M0 ((uint8_t) (0x01 << 4))
  TIM3 Output compare 2 mode [0] (in _TIM3_CCMR2)
- #define _TIM3_OC2M1 ((uint8_t) (0x01 << 5))
  TIM3 Output compare 2 mode [1] (in _TIM3_CCMR2)
- #define _TIM3_OC2M2 ((uint8_t) (0x01 << 6))
  TIM3 Output compare 2 mode [2] (in _TIM3_CCMR2)
- #define _TIM3_IC2PSC ((uint8_t) (0x03 << 2))
  TIM3 Input capture 2 prescaler [1:0] (in _TIM3_CCMR2)
- #define _TIM3_IC2PSC0 ((uint8_t) (0x01 << 2))
  TIM3 Input capture 2 prescaler [0] (in _TIM3_CCMR2)
- #define _TIM3_IC2PSC1 ((uint8_t) (0x01 << 3))
  TIM3 Input capture 2 prescaler [1] (in _TIM3_CCMR2)
- #define _TIM3_IC2F ((uint8_t) (0x0F << 4))

TIM3 Output compare 2 mode [3:0] (in _TIM3_CCMR2)

- #define _TIM3_IC2F0 ((uint8_t) (0x01 << 4))

  TIM3 Output compare 2 mode [0] (in _TIM3_CCMR2)
- #define _TIM3_IC2F1 ((uint8_t) (0x01 << 5))

  TIM3 Output compare 2 mode [1] (in _TIM3_CCMR2)
- #define _TIM3_IC2F2 ((uint8_t) (0x01 << 6))

  TIM3 Output compare 2 mode [2] (in _TIM3_CCMR2)
- #define _TIM3_IC2F3 ((uint8_t) (0x01 << 7))

  TIM3 Output compare 2 mode [3] (in _TIM3_CCMR2)
- #define _TIM3_CC1E ((uint8_t) (0x01 << 0))

  TIM3 Capture/compare 1 output enable [0] (in _TIM3_CCER1)
- #define _TIM3_CC1P ((uint8_t) (0x01 << 1))

  TIM3 Capture/compare 1 output polarity [0] (in _TIM3_CCER1)
- #define _TIM3_CC2E ((uint8_t) (0x01 << 4))

  TIM3 Capture/compare 2 output enable [0] (in _TIM3_CCER1)
- #define _TIM3_CC2P ((uint8_t) (0x01 << 5))

  TIM3 Capture/compare 2 output polarity [0] (in _TIM3_CCER1)
- #define _TIM3_PSC ((uint8_t) (0x0F << 0))

  TIM3 clock prescaler [3:0] (in _TIM3_PSCR)
- #define _TIM3_PSC0 ((uint8_t) (0x01 << 0))

  TIM3 clock prescaler [0] (in _TIM3_PSCR)
- #define _TIM3_PSC1 ((uint8_t) (0x01 << 1))

  TIM3 clock prescaler [1] (in _TIM3_PSCR)
- #define _TIM3_PSC2 ((uint8_t) (0x01 << 2))

  TIM3 clock prescaler [2] (in _TIM3_PSCR)
- #define _TIM3_PSC3 ((uint8_t) (0x01 << 3))

  TIM3 clock prescaler [3] (in _TIM3_PSCR)
- #define _TIM4 _SFR(TIM4_t, TIM4_AddressBase)

  TIM4 struct/bit access.
- #define _TIM4_CR _SFR(uint8_t, TIM4_AddressBase+0x00)

  TIM4 control register.
- #define _TIM4_IER _SFR(uint8_t, TIM4_AddressBase+0x01)

  TIM4 interrupt enable register.
- #define _TIM4_SR _SFR(uint8_t, TIM4_AddressBase+0x02)

  TIM4 status register.
- #define _TIM4_EGR _SFR(uint8_t, TIM4_AddressBase+0x03)

  TIM4 event generation register.
- #define _TIM4_CNTR _SFR(uint8_t, TIM4_AddressBase+0x04)

  TIM4 counter register.
- #define _TIM4_PSCR _SFR(uint8_t, TIM4_AddressBase+0x05)

  TIM4 clock prescaler register.
- #define _TIM4_ARR _SFR(uint8_t, TIM4_AddressBase+0x06)

  TIM4 auto-reload register.
- #define _TIM4_CR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 control register reset value.
- #define _TIM4_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM4 interrupt enable register reset value.
- #define _TIM4_SR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 status register reset value.
- #define _TIM4_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 event generation register reset value.

- #define _TIM4_CNTR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 counter register reset value.
- #define _TIM4_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 clock prescaler register reset value.
- #define _TIM4_ARR_RESET_VALUE ((uint8_t) 0xFF)

  TIM4 auto-reload register reset value.
- #define _TIM4_CEN ((uint8_t) (0x01 << 0))

  TIM4 Counter enable [0] (in _TIM4_CR)
- #define _TIM4_UDIS ((uint8_t) (0x01 << 1))

  TIM4 Update disable [0] (in _TIM4_CR)
- #define _TIM4_URS ((uint8_t) (0x01 << 2))

  TIM4 Update request source [0] (in _TIM4_CR)
- #define _TIM4_OPM ((uint8_t) (0x01 << 3))

  TIM4 One-pulse mode [0] (in _TIM4_CR)
- #define _TIM4_ARPE ((uint8_t) (0x01 << 7))

  TIM4 Auto-reload preload enable [0] (in _TIM4_CR)
- #define _TIM4_UIE ((uint8_t) (0x01 << 0))

  TIM4 Update interrupt enable [0] (in _TIM4_IER)
- #define _TIM4_UIF ((uint8_t) (0x01 << 0))

  TIM4 Update interrupt flag [0] (in _TIM4_SR)
- #define _TIM4_UG ((uint8_t) (0x01 << 0))

  TIM4 Update generation [0] (in _TIM4_EGR)
- #define _TIM4_PSC ((uint8_t) (0x07 << 0))

  TIM4 clock prescaler [2:0] (in _TIM4_PSCR)
- #define _TIM4_PSC0 ((uint8_t) (0x01 << 0))

  TIM4 clock prescaler [0] (in _TIM4_PSCR)
- #define _TIM4_PSC1 ((uint8_t) (0x01 << 1))

  TIM4 clock prescaler [1] (in _TIM4_PSCR)
- #define _TIM4_PSC2 ((uint8_t) (0x01 << 2))

  TIM4 clock prescaler [2] (in _TIM4_PSCR)
- #define _TIM5 _SFR(TIM5_t, TIM5_AddressBase)

  TIM5 struct/bit access.
- #define _TIM5_CR1 _SFR(uint8_t, TIM5_AddressBase+0x00)

  TIM5 control register 1.
- #define _TIM5_CR2 _SFR(uint8_t, TIM5_AddressBase+0x01)

  TIM5 control register 2.
- #define _TIM5_SMCR _SFR(uint8_t, TIM5_AddressBase+0x02)

  TIM5 Slave mode control register.
- #define _TIM5_IER _SFR(uint8_t, TIM5_AddressBase+0x03)

  TIM5 interrupt enable register.
- #define _TIM5_SR1 _SFR(uint8_t, TIM5_AddressBase+0x04)

  TIM5 status register 1.
- #define _TIM5_SR2 _SFR(uint8_t, TIM5_AddressBase+0x05)

  TIM5 status register 2.
- #define _TIM5_EGR _SFR(uint8_t, TIM5_AddressBase+0x06)

  TIM5 Event generation register.
- #define _TIM5_CCMR1 _SFR(uint8_t, TIM5_AddressBase+0x07)

  TIM5 Capture/compare mode register 1.
- #define _TIM5_CCMR2 _SFR(uint8_t, TIM5_AddressBase+0x08)

  TIM5 Capture/compare mode register 2.
- #define _TIM5_CCMR3 _SFR(uint8_t, TIM5_AddressBase+0x09)

TIM5 Capture/compare mode register 3.

- #define _TIM5_CCER1 _SFR(uint8_t, TIM5_AddressBase+0x0A)

  TIM5 Capture/compare enable register 1.
- #define _TIM5_CCER2 _SFR(uint8_t, TIM5_AddressBase+0x0B)

  TIM5 Capture/compare enable register 2.
- #define _TIM5_CNTRH _SFR(uint8_t, TIM5_AddressBase+0x0C)

  TIM5 counter register high byte.
- #define _TIM5_CNTRL _SFR(uint8_t, TIM5_AddressBase+0x0D)

  TIM5 counter register low byte.
- #define _TIM5_PSCR _SFR(uint8_t, TIM5_AddressBase+0x0E)

  TIM5 clock prescaler register.
- #define _TIM5_ARRH _SFR(uint8_t, TIM5_AddressBase+0x0F)

  TIM5 auto-reload register high byte.
- #define _TIM5_ARRL _SFR(uint8_t, TIM5_AddressBase+0x10)

  TIM5 auto-reload register low byte.
- #define _TIM5_CCR1H _SFR(uint8_t, TIM5_AddressBase+0x11)

  TIM5 16-bit capture/compare value 1 high byte.
- #define _TIM5_CCR1L _SFR(uint8_t, TIM5_AddressBase+0x12)

  TIM5 16-bit capture/compare value 1 low byte.
- #define _TIM5_CCR2H _SFR(uint8_t, TIM5_AddressBase+0x13)

  TIM5 16-bit capture/compare value 2 high byte.
- #define _TIM5_CCR2L _SFR(uint8_t, TIM5_AddressBase+0x14)

  TIM5 16-bit capture/compare value 2 low byte.
- #define _TIM5_CCR3H _SFR(uint8_t, TIM5_AddressBase+0x15)

  TIM5 16-bit capture/compare value 3 high byte.
- #define _TIM5_CCR3L _SFR(uint8_t, TIM5_AddressBase+0x16)

  TIM5 16-bit capture/compare value 3 low byte.
- #define _TIM5_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM5 control register 1 reset value.
- #define _TIM5_CR2_RESET_VALUE ((uint8_t) 0x00)

  TIM5 control register 2 reset value.
- #define _TIM5_SMCR_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Slave mode control register reset value.
- #define _TIM5_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM5 interrupt enable register reset value.
- #define _TIM5_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM5 status register 1 reset value.
- #define _TIM5_SR2_RESET_VALUE ((uint8_t) 0x00)

  TIM5 status register 2 reset value.
- #define _TIM5_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Event generation register reset value.
- #define _TIM5_CCMR1_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Capture/compare mode register 1 reset value.
- #define _TIM5_CCMR2_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Capture/compare mode register 2 reset value.
- #define _TIM5_CCMR3_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Capture/compare mode register 3 reset value.
- #define _TIM5_CCER1_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Capture/compare enable register 1 reset value.
- #define _TIM5_CCER2_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Capture/compare enable register 2 reset value.

- #define _TIM5_CNTRH_RESET_VALUE ((uint8_t) 0x00)

  TIM5 counter register high byte reset value.
- #define _TIM5_CNTRL_RESET_VALUE ((uint8_t) 0x00)

  TIM5 counter register low byte reset value.
- #define _TIM5_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM5 clock prescaler register reset value.
- #define _TIM5_ARRH_RESET_VALUE ((uint8_t) 0xFF)

  TIM5 auto-reload register high byte reset value.
- #define _TIM5_ARRL_RESET_VALUE ((uint8_t) 0xFF)

  TIM5 auto-reload register low byte reset value.
- #define _TIM5_CCR1H_RESET_VALUE ((uint8_t) 0x00)

  TIM5 16-bit capture/compare value 1 high byte reset value.
- #define _TIM5_CCR1L_RESET_VALUE ((uint8_t) 0x00)

  TIM5 16-bit capture/compare value 1 low byte reset value.
- #define _TIM5_CCR2H_RESET_VALUE ((uint8_t) 0x00)

  TIM5 16-bit capture/compare value 2 high byte reset value.
- #define _TIM5_CCR2L_RESET_VALUE ((uint8_t) 0x00)

  TIM5 16-bit capture/compare value 2 low byte reset value.
- #define _TIM5_CCR3H_RESET_VALUE ((uint8_t) 0x00)

  TIM5 16-bit capture/compare value 3 high byte reset value.
- #define _TIM5_CCR3L_RESET_VALUE ((uint8_t) 0x00)

  TIM5 16-bit capture/compare value 3 low byte reset value.
- #define _TIM5_CEN ((uint8_t) (0x01 << 0))

  TIM5 Counter enable [0] (in _TIM5_CR1)
- #define _TIM5_UDIS ((uint8_t) (0x01 << 1))

  TIM5 Update disable [0] (in _TIM5_CR1)
- #define _TIM5_URS ((uint8_t) (0x01 << 2))

  TIM5 Update request source [0] (in _TIM5_CR1)
- #define _TIM5_OPM ((uint8_t) (0x01 << 3))

  TIM5 One-pulse mode [0] (in _TIM5_CR1)
- #define _TIM5_ARPE ((uint8_t) (0x01 << 7))

  TIM5 Auto-reload preload enable [0] (in _TIM5_CR1)
- #define _TIM5_CCPC ((uint8_t) (0x01 << 0))

  TIM5 Capture/compare preloaded control [0] (in _TIM5_CR2)
- #define _TIM5_COMS ((uint8_t) (0x01 << 2))

  TIM5 Capture/compare control update selection [0] (in _TIM5_CR2)
- #define _TIM5_MMS ((uint8_t) (0x07 << 4))

  TIM5 Master mode selection [2:0] (in _TIM5_CR2)
- #define _TIM5_MMS0 ((uint8_t) (0x01 << 4))

  TIM5 Master mode selection [0] (in _TIM5_CR2)
- #define _TIM5_MMS1 ((uint8_t) (0x01 << 5))

  TIM5 Master mode selection [1] (in _TIM5_CR2)
- #define _TIM5_MMS2 ((uint8_t) (0x01 << 6))

  TIM5 Master mode selection [2] (in _TIM5_CR2)
- #define _TIM5_SMS ((uint8_t) (0x07 << 0))

  TIM5 Clock/trigger/slave mode selection [2:0] (in _TIM5_SMCR)
- #define _TIM5_SMS0 ((uint8_t) (0x01 << 0))

  TIM5 Clock/trigger/slave mode selection [0] (in _TIM5_SMCR)
- #define _TIM5_SMS1 ((uint8_t) (0x01 << 1))

  TIM5 Clock/trigger/slave mode selection [1] (in _TIM5_SMCR)
- #define _TIM5_SMS2 ((uint8_t) (0x01 << 2))

TIM5 Clock/trigger/slave mode selection [2] (in _TIM5_SMCR)

- #define _TIM5_TS ((uint8_t) (0x07 << 4))

  TIM5 Trigger selection [2:0] (in _TIM5_SMCR)
- #define _TIM5_TS0 ((uint8_t) (0x01 << 4))

  TIM5 Trigger selection [0] (in _TIM5_SMCR)
- #define _TIM5_TS1 ((uint8_t) (0x01 << 5))

  TIM5 Trigger selection [1] (in _TIM5_SMCR)
- #define _TIM5_TS2 ((uint8_t) (0x01 << 6))

  TIM5 Trigger selection [2] (in _TIM5_SMCR)
- #define _TIM5_MSM ((uint8_t) (0x01 << 7))

  TIM5 Master/slave mode [0] (in _TIM5_SMCR)
- #define _TIM5_UIE ((uint8_t) (0x01 << 0))

  TIM5 Update interrupt enable [0] (in _TIM5_IER)
- #define _TIM5_CC1IE ((uint8_t) (0x01 << 1))

  TIM5 Capture/compare 1 interrupt enable [0] (in _TIM5_IER)
- #define _TIM5_CC2IE ((uint8_t) (0x01 << 2))

  TIM5 Capture/compare 2 interrupt enable [0] (in _TIM5_IER)
- #define _TIM5_CC3IE ((uint8_t) (0x01 << 3))

  TIM5 Capture/compare 3 interrupt enable [0] (in _TIM5_IER)
- #define _TIM5_TIE ((uint8_t) (0x01 << 6))

  TIM5 Trigger interrupt enable [0] (in _TIM5_IER)
- #define _TIM5_UIF ((uint8_t) (0x01 << 0))

  TIM5 Update interrupt flag [0] (in _TIM5_SR1)
- #define _TIM5_CC1IF ((uint8_t) (0x01 << 1))

  TIM5 Capture/compare 1 interrupt flag [0] (in _TIM5_SR1)
- #define _TIM5_CC2IF ((uint8_t) (0x01 << 2))

  TIM5 Capture/compare 2 interrupt flag [0] (in _TIM5_SR1)
- #define _TIM5_CC3IF ((uint8_t) (0x01 << 3))

  TIM5 Capture/compare 3 interrupt flag [0] (in _TIM5_SR1)
- #define _TIM5_TIF ((uint8_t) (0x01 << 6))

  TIM5 Trigger interrupt flag [0] (in _TIM5_SR1)
- #define _TIM5_CC1OF ((uint8_t) (0x01 << 1))

  TIM5 Capture/compare 1 overcapture flag [0] (in _TIM5_SR2)
- #define _TIM5_CC2OF ((uint8_t) (0x01 << 2))

  TIM5 Capture/compare 2 overcapture flag [0] (in _TIM5_SR2)
- #define _TIM5_CC3OF ((uint8_t) (0x01 << 3))

  TIM5 Capture/compare 3 overcapture flag [0] (in _TIM5_SR2)
- #define _TIM5_UG ((uint8_t) (0x01 << 0))

  TIM5 Update generation [0] (in _TIM5_EGR)
- #define _TIM5_CC1G ((uint8_t) (0x01 << 1))

  TIM5 Capture/compare 1 generation [0] (in _TIM5_EGR)
- #define _TIM5_CC2G ((uint8_t) (0x01 << 2))

  TIM5 Capture/compare 2 generation [0] (in _TIM5_EGR)
- #define _TIM5_CC3G ((uint8_t) (0x01 << 3))

  TIM5 Capture/compare 3 generation [0] (in _TIM5_EGR)
- #define _TIM5_TG ((uint8_t) (0x01 << 6))

  TIM5 Trigger generation [0] (in _TIM5_EGR)
- #define _TIM5_CC1S ((uint8_t) (0x03 << 0))

  TIM5 Compare 1 selection [1:0] (in _TIM5_CCMR1)
- #define _TIM5_CC1S0 ((uint8_t) (0x01 << 0))

  TIM5 Compare 1 selection [0] (in _TIM5_CCMR1)

- #define _TIM5_CC1S1 ((uint8_t) (0x01 << 1))
  TIM5 Compare 1 selection [1] (in _TIM5_CCMR1)
- #define _TIM5_OC1PE ((uint8_t) (0x01 << 3))
  TIM5 Output compare 1 preload enable [0] (in _TIM5_CCMR1)
- #define _TIM5_OC1M ((uint8_t) (0x07 << 4))
  TIM5 Output compare 1 mode [2:0] (in _TIM5_CCMR1)
- #define _TIM5_OC1M0 ((uint8_t) (0x01 << 4))
  TIM5 Output compare 1 mode [0] (in _TIM5_CCMR1)
- #define _TIM5_OC1M1 ((uint8_t) (0x01 << 5))
  TIM5 Output compare 1 mode [1] (in _TIM5_CCMR1)
- #define _TIM5_OC1M2 ((uint8_t) (0x01 << 6))
  TIM5 Output compare 1 mode [2] (in _TIM5_CCMR1)
- #define _TIM5_IC1PSC ((uint8_t) (0x03 << 2))
  TIM5 Input capture 1 prescaler [1:0] (in _TIM5_CCMR1)
- #define _TIM5_IC1PSC0 ((uint8_t) (0x01 << 2))
  TIM5 Input capture 1 prescaler [0] (in _TIM5_CCMR1)
- #define _TIM5_IC1PSC1 ((uint8_t) (0x01 << 3))
  TIM5 Input capture 1 prescaler [1] (in _TIM5_CCMR1)
- #define _TIM5_IC1F ((uint8_t) (0x0F << 4))
  TIM5 Output compare 1 mode [3:0] (in _TIM5_CCMR1)
- #define _TIM5_IC1F0 ((uint8_t) (0x01 << 4))
  TIM5 Input capture 1 filter [0] (in _TIM5_CCMR1)
- #define _TIM5_IC1F1 ((uint8_t) (0x01 << 5))
  TIM5 Input capture 1 filter [1] (in _TIM5_CCMR1)
- #define _TIM5_IC1F2 ((uint8_t) (0x01 << 6))
  TIM5 Input capture 1 filter [2] (in _TIM5_CCMR1)
- #define _TIM5_IC1F3 ((uint8_t) (0x01 << 7))
  TIM5 Input capture 1 filter [3] (in _TIM5_CCMR1)
- #define _TIM5_CC2S ((uint8_t) (0x03 << 0))
  TIM5 Compare 2 selection [1:0] (in _TIM5_CCMR2)
- #define _TIM5_CC2S0 ((uint8_t) (0x01 << 0))
  TIM5 Compare 2 selection [0] (in _TIM5_CCMR2)
- #define _TIM5_CC2S1 ((uint8_t) (0x01 << 1))
  TIM5 Compare 2 selection [1] (in _TIM5_CCMR2)
- #define _TIM5_OC2PE ((uint8_t) (0x01 << 3))
  TIM5 Output compare 2 preload enable [0] (in _TIM5_CCMR2)
- #define _TIM5_OC2M ((uint8_t) (0x07 << 4))
  TIM5 Output compare 2 mode [2:0] (in _TIM5_CCMR2)
- #define _TIM5_OC2M0 ((uint8_t) (0x01 << 4))
  TIM5 Output compare 2 mode [0] (in _TIM5_CCMR2)
- #define _TIM5_OC2M1 ((uint8_t) (0x01 << 5))
  TIM5 Output compare 2 mode [1] (in _TIM5_CCMR2)
- #define _TIM5_OC2M2 ((uint8_t) (0x01 << 6))
  TIM5 Output compare 2 mode [2] (in _TIM5_CCMR2)
- #define _TIM5_IC2PSC ((uint8_t) (0x03 << 2))
  TIM5 Input capture 2 prescaler [1:0] (in _TIM5_CCMR2)
- #define _TIM5_IC2PSC0 ((uint8_t) (0x01 << 2))
  TIM5 Input capture 2 prescaler [0] (in _TIM5_CCMR2)
- #define _TIM5_IC2PSC1 ((uint8_t) (0x01 << 3))
  TIM5 Input capture 2 prescaler [1] (in _TIM5_CCMR2)
- #define _TIM5_IC2F ((uint8_t) (0x0F << 4))

TIM5 Output compare 2 mode [3:0] (in _TIM5_CCMR2)

- #define _TIM5_IC2F0 ((uint8_t) (0x01 << 4))

  TIM5 Input capture 2 filter [0] (in _TIM5_CCMR2)
- #define _TIM5_IC2F1 ((uint8_t) (0x01 << 5))

  TIM5 Input capture 2 filter [1] (in _TIM5_CCMR2)
- #define _TIM5_IC2F2 ((uint8_t) (0x01 << 6))

  TIM5 Input capture 2 filter [2] (in _TIM5_CCMR2)
- #define _TIM5_IC2F3 ((uint8_t) (0x01 << 7))

  TIM5 Input capture 2 filter [3] (in _TIM5_CCMR2)
- #define _TIM5_CC3S ((uint8_t) (0x03 << 0))

  TIM5 Compare 3 selection [1:0] (in _TIM5_CCMR3)
- #define _TIM5_CC3S0 ((uint8_t) (0x01 << 0))

  TIM5 Compare 3 selection [0] (in _TIM5_CCMR3)
- #define _TIM5_CC3S1 ((uint8_t) (0x01 << 1))

  TIM5 Compare 3 selection [1] (in _TIM5_CCMR3)
- #define _TIM5_OC3PE ((uint8_t) (0x01 << 3))

  TIM5 Output compare 3 preload enable [0] (in _TIM5_CCMR3)
- #define _TIM5_OC3M ((uint8_t) (0x07 << 4))

  TIM5 Output compare 3 mode [2:0] (in _TIM5_CCMR3)
- #define _TIM5_OC3M0 ((uint8_t) (0x01 << 4))

  TIM5 Output compare 3 mode [0] (in _TIM5_CCMR3)
- #define _TIM5_OC3M1 ((uint8_t) (0x01 << 5))

  TIM5 Output compare 3 mode [1] (in _TIM5_CCMR3)
- #define _TIM5_OC3M2 ((uint8_t) (0x01 << 6))

  TIM5 Output compare 3 mode [2] (in _TIM5_CCMR3)
- #define _TIM5_IC3PSC ((uint8_t) (0x03 << 2))

  TIM5 Input capture 3 prescaler [1:0] (in _TIM5_CCMR3)
- #define _TIM5_IC3PSC0 ((uint8_t) (0x01 << 2))

  TIM5 Input capture 3 prescaler [0] (in _TIM5_CCMR3)
- #define _TIM5_IC3PSC1 ((uint8_t) (0x01 << 3))

  TIM5 Input capture 3 prescaler [1] (in _TIM5_CCMR3)
- #define _TIM5_IC3F ((uint8_t) (0x0F << 4))

  TIM5 Output compare 3 mode [3:0] (in _TIM5_CCMR3)
- #define _TIM5_IC3F0 ((uint8_t) (0x01 << 4))

  TIM5 Input capture 3 filter [0] (in _TIM5_CCMR3)
- #define _TIM5_IC3F1 ((uint8_t) (0x01 << 5))

  TIM5 Input capture 3 filter [1] (in _TIM5_CCMR3)
- #define _TIM5_IC3F2 ((uint8_t) (0x01 << 6))

  TIM5 Input capture 3 filter [2] (in _TIM5_CCMR3)
- #define _TIM5_IC3F3 ((uint8_t) (0x01 << 7))

  TIM5 Input capture 3 filter [3] (in _TIM5_CCMR3)
- #define _TIM5_CC1E ((uint8_t) (0x01 << 0))

  TIM5 Capture/compare 1 output enable [0] (in _TIM5_CCER1)
- #define _TIM5_CC1P ((uint8_t) (0x01 << 1))

  TIM5 Capture/compare 1 output polarity [0] (in _TIM5_CCER1)
- #define _TIM5_CC2E ((uint8_t) (0x01 << 4))

  TIM5 Capture/compare 2 output enable [0] (in _TIM5_CCER1)
- #define _TIM5_CC2P ((uint8_t) (0x01 << 5))

  TIM5 Capture/compare 2 output polarity [0] (in _TIM5_CCER1)
- #define _TIM5_CC3E ((uint8_t) (0x01 << 0))

  TIM5 Capture/compare 3 output enable [0] (in _TIM5_CCER2)

- #define _TIM5_CC3P ((uint8_t) (0x01 << 1))

  TIM5 Capture/compare 3 output polarity [0] (in _TIM5_CCER2)
- #define _TIM5_PSC ((uint8_t) (0x0F << 0))

  TIM5 clock prescaler [3:0] (in _TIM5_PSCR)
- #define _TIM5_PSC0 ((uint8_t) (0x01 << 0))

  TIM5 clock prescaler [0] (in _TIM5_PSCR)
- #define _TIM5_PSC1 ((uint8_t) (0x01 << 1))

  TIM5 clock prescaler [1] (in _TIM5_PSCR)
- #define _TIM5_PSC2 ((uint8_t) (0x01 << 2))

  TIM5 clock prescaler [2] (in _TIM5_PSCR)
- #define _TIM5_PSC3 ((uint8_t) (0x01 << 3))

  TIM5 clock prescaler [3] (in _TIM5_PSCR)
- #define _TIM6 _SFR(TIM6_t, TIM6_AddressBase)

  TIM6 struct/bit access.
- #define _TIM6_CR _SFR(uint8_t, TIM6_AddressBase+0x00)

  TIM6 control register.
- #define _TIM6_IER _SFR(uint8_t, TIM6_AddressBase+0x01)

  TIM6 interrupt enable register.
- #define _TIM6_SR _SFR(uint8_t, TIM6_AddressBase+0x02)

  TIM6 status register.
- #define _TIM6_EGR _SFR(uint8_t, TIM6_AddressBase+0x03)

  TIM6 event generation register.
- #define _TIM6_CNTR _SFR(uint8_t, TIM6_AddressBase+0x04)

  TIM6 counter register.
- #define _TIM6_PSCR _SFR(uint8_t, TIM6_AddressBase+0x05)

  TIM6 clock prescaler register.
- #define _TIM6_ARR _SFR(uint8_t, TIM6_AddressBase+0x06)

  TIM6 auto-reload register.
- #define _TIM6_CR_RESET_VALUE ((uint8_t) 0x00)

  TIM6 control register reset value.
- #define _TIM6_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM6 interrupt enable register reset value.
- #define _TIM6_SR_RESET_VALUE ((uint8_t) 0x00)

  TIM6 status register reset value.
- #define _TIM6_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM6 event generation register reset value.
- #define _TIM6_CNTR_RESET_VALUE ((uint8_t) 0x00)

  TIM6 counter register reset value.
- #define _TIM6_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM6 clock prescaler register reset value.
- #define _TIM6_ARR_RESET_VALUE ((uint8_t) 0xFF)

  TIM6 auto-reload register reset value.
- #define _TIM6_CEN ((uint8_t) (0x01 << 0))

  TIM6 Counter enable [0] (in _TIM6_CR1)
- #define _TIM6_UDIS ((uint8_t) (0x01 << 1))

  TIM6 Update disable [0] (in _TIM6_CR1)
- #define _TIM6_URS ((uint8_t) (0x01 << 2))

  TIM6 Update request source [0] (in _TIM6_CR1)
- #define _TIM6_OPM ((uint8_t) (0x01 << 3))

  TIM6 One-pulse mode [0] (in _TIM6_CR1)
- #define _TIM6_ARPE ((uint8_t) (0x01 << 7))

TIM6 Auto-reload preload enable [0] (in _TIM6_CR1)

- #define _TIM6_MMS ((uint8_t) (0x07 << 4))

  TIM6 Master mode selection [2:0] (in _TIM6_CR2)
- #define _TIM6_MMS0 ((uint8_t) (0x01 << 4))

  TIM6 Master mode selection [0] (in _TIM6_CR2)
- #define _TIM6_MMS1 ((uint8_t) (0x01 << 5))

  TIM6 Master mode selection [1] (in _TIM6_CR2)
- #define _TIM6_MMS2 ((uint8_t) (0x01 << 6))

  TIM6 Master mode selection [2] (in _TIM6_CR2)
- #define _TIM6_SMS ((uint8_t) (0x07 << 0))

  TIM6 Clock/trigger/slave mode selection [2:0] (in _TIM6_SMCR)
- #define _TIM6_SMS0 ((uint8_t) (0x01 << 0))

  TIM6 Clock/trigger/slave mode selection [0] (in _TIM6_SMCR)
- #define _TIM6_SMS1 ((uint8_t) (0x01 << 1))

  TIM6 Clock/trigger/slave mode selection [1] (in _TIM6_SMCR)
- #define _TIM6_SMS2 ((uint8_t) (0x01 << 2))

  TIM6 Clock/trigger/slave mode selection [2] (in _TIM6_SMCR)
- #define _TIM6_TS ((uint8_t) (0x07 << 4))

  TIM6 Trigger selection [2:0] (in _TIM6_SMCR)
- #define _TIM6_TS0 ((uint8_t) (0x01 << 4))

  TIM6 Trigger selection [0] (in _TIM6_SMCR)
- #define _TIM6_TS1 ((uint8_t) (0x01 << 5))

  TIM6 Trigger selection [1] (in _TIM6_SMCR)
- #define _TIM6_TS2 ((uint8_t) (0x01 << 6))

  TIM6 Trigger selection [2] (in _TIM6_SMCR)
- #define _TIM6_UIE ((uint8_t) (0x01 << 0))

  TIM6 Update interrupt enable [0] (in _TIM6_IER)
- #define _TIM6_UIF ((uint8_t) (0x01 << 0))

  TIM6 Update interrupt flag [0] (in _TIM6_SR)
- #define _TIM6_UG ((uint8_t) (0x01 << 0))

  TIM6 Update generation [0] (in _TIM6_EGR)
- #define _TIM6_PSC ((uint8_t) (0x07 << 0))

  TIM6 clock prescaler [2:0] (in _TIM6_PSCR)
- #define _TIM6_PSC0 ((uint8_t) (0x01 << 0))

  TIM6 clock prescaler [0] (in _TIM6_PSCR)
- #define _TIM6_PSC1 ((uint8_t) (0x01 << 1))

  TIM6 clock prescaler [1] (in _TIM6_PSCR)
- #define _TIM6_PSC2 ((uint8_t) (0x01 << 2))

  TIM6 clock prescaler [2] (in _TIM6_PSCR)
- #define _ADC1 _SFR(ADC1_t, ADC1_AddressBase)

  ADC1 struct/bit access.
- #define _ADC1_DB0RH _SFR(uint8_t, ADC1_AddressBase+0x00)

  ADC1 10-bit Data Buffer Register 0.
- #define _ADC1_DB0RL _SFR(uint8_t, ADC1_AddressBase+0x01)

  ADC1 10-bit Data Buffer Register 0.
- #define _ADC1_DB1RH _SFR(uint8_t, ADC1_AddressBase+0x02)

  ADC1 10-bit Data Buffer Register 1.
- #define _ADC1_DB1RL _SFR(uint8_t, ADC1_AddressBase+0x03)

  ADC1 10-bit Data Buffer Register 1.
- #define _ADC1_DB2RH _SFR(uint8_t, ADC1_AddressBase+0x04)

  ADC1 10-bit Data Buffer Register 2.

- #define _ADC1_DB2RL _SFR(uint8_t, ADC1_AddressBase+0x05)

  ADC1 10-bit Data Buffer Register 2.
- #define _ADC1_DB3RH _SFR(uint8_t, ADC1_AddressBase+0x06)

  ADC1 10-bit Data Buffer Register 3.
- #define _ADC1_DB3RL _SFR(uint8_t, ADC1_AddressBase+0x07)

  ADC1 10-bit Data Buffer Register 3.
- #define _ADC1_DB4RH _SFR(uint8_t, ADC1_AddressBase+0x08)

  ADC1 10-bit Data Buffer Register 4.
- #define _ADC1_DB4RL _SFR(uint8_t, ADC1_AddressBase+0x09)

  ADC1 10-bit Data Buffer Register 4.
- #define _ADC1_DB5RH _SFR(uint8_t, ADC1_AddressBase+0x0A)

  ADC1 10-bit Data Buffer Register 5.
- #define _ADC1_DB5RL _SFR(uint8_t, ADC1_AddressBase+0x0B)

  ADC1 10-bit Data Buffer Register 5.
- #define _ADC1_DB6RH _SFR(uint8_t, ADC1_AddressBase+0x0C)

  ADC1 10-bit Data Buffer Register 6.
- #define _ADC1_DB6RL _SFR(uint8_t, ADC1_AddressBase+0x0D)

  ADC1 10-bit Data Buffer Register 6.
- #define _ADC1_DB7RH _SFR(uint8_t, ADC1_AddressBase+0x0E)

  ADC1 10-bit Data Buffer Register 7.
- #define _ADC1_DB7RL _SFR(uint8_t, ADC1_AddressBase+0x0F)

  ADC1 10-bit Data Buffer Register 7.
- #define _ADC1_DB8RH _SFR(uint8_t, ADC1_AddressBase+0x10)

  ADC1 10-bit Data Buffer Register 8.
- #define _ADC1_DB8RL _SFR(uint8_t, ADC1_AddressBase+0x11)

  ADC1 10-bit Data Buffer Register 8.
- #define _ADC1_DB9RH _SFR(uint8_t, ADC1_AddressBase+0x12)

  ADC1 10-bit Data Buffer Register 9.
- #define _ADC1_DB9RL _SFR(uint8_t, ADC1_AddressBase+0x13)

  ADC1 10-bit Data Buffer Register 9.
- #define _ADC1_CSR _SFR(uint8_t, ADC1_AddressBase+0x20)

  ADC1 control/status register.
- #define _ADC1_CR1 _SFR(uint8_t, ADC1_AddressBase+0x21)

  ADC1 Configuration Register 1.
- #define _ADC1_CR2 _SFR(uint8_t, ADC1_AddressBase+0x22)

  ADC1 Configuration Register 2.
- #define _ADC1_CR3 _SFR(uint8_t, ADC1_AddressBase+0x23)

  ADC1 Configuration Register 3.
- #define _ADC1_DRH _SFR(uint8_t, ADC1_AddressBase+0x24)

  ADC1 (unbuffered) 10-bit measurement result.
- #define _ADC1_DRL _SFR(uint8_t, ADC1_AddressBase+0x25)

  ADC1 (unbuffered) 10-bit measurement result.
- #define _ADC1_TDRH _SFR(uint8_t, ADC1_AddressBase+0x26)

  ADC1 Schmitt trigger disable register.
- #define _ADC1_TDRL _SFR(uint8_t, ADC1_AddressBase+0x27)

  ADC1 Schmitt trigger disable register.
- #define _ADC1_HTRH _SFR(uint8_t, ADC1_AddressBase+0x28)

  ADC1 watchdog high threshold register.
- #define _ADC1_HTRL _SFR(uint8_t, ADC1_AddressBase+0x29)

  ADC1 watchdog high threshold register.
- #define _ADC1_LTRH _SFR(uint8_t, ADC1_AddressBase+0x2A)

ADC1 watchdog low threshold register.

- #define _ADC1_LTRL _SFR(uint8_t, ADC1_AddressBase+0x2B)

  ADC1 watchdog low threshold register.
- #define _ADC1_AWSRH _SFR(uint8_t, ADC1_AddressBase+0x2C)

  ADC1 watchdog status register.
- #define _ADC1_AWSRL _SFR(uint8_t, ADC1_AddressBase+0x2D)

  ADC1 watchdog status register.
- #define _ADC1_AWCRH _SFR(uint8_t, ADC1_AddressBase+0x2E)

  ADC1 watchdog control register.
- #define _ADC1_AWCRL _SFR(uint8_t, ADC1_AddressBase+0x2F)

  ADC1 watchdog control register.
- #define _ADC1_CSR_RESET_VALUE ((uint8_t) 0x00)

  ADC1 control/status register reset value.
- #define _ADC1_CR1_RESET_VALUE ((uint8_t) 0x00)

  ADC1 Configuration Register 1 reset value.
- #define _ADC1_CR2_RESET_VALUE ((uint8_t) 0x00)

  ADC1 Configuration Register 2 reset value.
- #define _ADC1_CR3_RESET_VALUE ((uint8_t) 0x00)

  ADC1 Configuration Register 3 reset value.
- #define _ADC1_TDRH_RESET_VALUE ((uint8_t) 0x00)

  ADC1 Schmitt trigger disable register reset value.
- #define _ADC1_TDRL_RESET_VALUE ((uint8_t) 0x00)

  ADC1 Schmitt trigger disable register reset value.
- #define _ADC1_HTRH_RESET_VALUE ((uint8_t) 0xFF)

  ADC1 watchdog high threshold register reset value.
- #define _ADC1_HTRL_RESET_VALUE ((uint8_t) 0x03)

  ADC1 watchdog high threshold register reset value.
- #define _ADC1_LTRH_RESET_VALUE ((uint8_t) 0x00)

  ADC1 watchdog low threshold register reset value.
- #define _ADC1_LTRL_RESET_VALUE ((uint8_t) 0x00)

  ADC1 watchdog low threshold register reset value.
- #define _ADC1_AWCRH_RESET_VALUE ((uint8_t) 0x00)

  ADC1 watchdog control register reset value.
- #define _ADC1_AWCRL_RESET_VALUE ((uint8_t) 0x00)

  ADC1 watchdog control register reset value.
- #define _ADC1_CH ((uint8_t) (0x0F << 0))

  ADC1 Channel selection bits [3:0] (in _ADC1_CSR)
- #define _ADC1_CH0 ((uint8_t) (0x01 << 0))

  ADC1 Channel selection bits [0] (in _ADC1_CSR)
- #define _ADC1_CH1 ((uint8_t) (0x01 << 1))

  ADC1 Channel selection bits [1] (in _ADC1_CSR)
- #define _ADC1_CH2 ((uint8_t) (0x01 << 2))

  ADC1 Channel selection bits [2] (in _ADC1_CSR)
- #define _ADC1_CH3 ((uint8_t) (0x01 << 3))

  ADC1 Channel selection bits [3] (in _ADC1_CSR)
- #define _ADC1_AWDIE ((uint8_t) (0x01 << 4))

  ADC1 Analog watchdog interrupt enable [0] (in _ADC1_CSR)
- #define _ADC1_EOCIE ((uint8_t) (0x01 << 5))

  ADC1 Interrupt enable for EOC [0] (in _ADC1_CSR)
- #define _ADC1_AWD ((uint8_t) (0x01 << 6))

  ADC1 Analog Watchdog flag [0] (in _ADC1_CSR)

- #define _ADC1_EOC ((uint8_t) (0x01 << 7))
  ADC1 End of conversion [0] (in _ADC1_CSR)
- #define _ADC1_ADON ((uint8_t) (0x01 << 0))
  ADC1 Conversion on/off [0] (in _ADC1_CR1)
- #define _ADC1_CONT ((uint8_t) (0x01 << 1))
  ADC1 Continuous conversion [0] (in _ADC1_CR1)
- #define _ADC1_SPSEL ((uint8_t) (0x07 << 4)
  ADC1 clock prescaler selection [2:0] (in _ADC1_CR1)
- #define _ADC1_SPSEL0 ((uint8_t) (0x01 << 4)
  ADC1 clock prescaler selection [0] (in _ADC1_CR1)
- #define _ADC1_SPSEL1 ((uint8_t) (0x01 << 5)
  ADC1 clock prescaler selection [1] (in _ADC1_CR1)
- #define _ADC1_SPSEL2 ((uint8_t) (0x01 << 6)
  ADC1 clock prescaler selection [2] (in _ADC1_CR1)
- #define _ADC1_SCAN ((uint8_t) (0x01 << 1))
  ADC1 Scan mode enable [0] (in _ADC1_CR2)
- #define _ADC1_ALIGN ((uint8_t) (0x01 << 3))
  ADC1 Data alignment [0] (in _ADC1_CR2)
- #define _ADC1_EXTSEL ((uint8_t) (0x03 << 4)
  ADC1 External event selection [1:0] (in _ADC1_CR2)
- #define _ADC1_EXTSEL0 ((uint8_t) (0x01 << 4)
  ADC1 External event selection [0] (in _ADC1_CR2)
- #define _ADC1_EXTSEL1 ((uint8_t) (0x01 << 5)
  ADC1 External event selection [1] (in _ADC1_CR2)
- #define _ADC1_EXTTRIG ((uint8_t) (0x01 << 6))
  ADC1 External trigger enable [0] (in _ADC1_CR2)
- #define _ADC1_OVR ((uint8_t) (0x01 << 6))
  ADC1 Overrun flag [0] (in _ADC1_CR3)
- #define _ADC1_DBUF ((uint8_t) (0x01 << 7))
  ADC1 Data buffer enable [0] (in _ADC1_CR3)
- #define _ADC2 _SFR(ADC2_t, ADC2_AddressBase)
  ADC2 struct/bit access.
- #define _ADC2_CSR _SFR(uint8_t, ADC2_AddressBase+0x00)
  ADC2 control/status register.
- #define _ADC2_CR1 _SFR(uint8_t, ADC2_AddressBase+0x01)
  ADC2 Configuration Register 1.
- #define _ADC2_CR2 _SFR(uint8_t, ADC2_AddressBase+0x02)
  ADC2 Configuration Register 2.
- #define _ADC2_DRH _SFR(uint8_t, ADC2_AddressBase+0x04)
  ADC2 (unbuffered) 10-bit measurement result.
- #define _ADC2_DRL _SFR(uint8_t, ADC2_AddressBase+0x05)
  ADC2 (unbuffered) 10-bit measurement result.
- #define _ADC2_TDRH _SFR(uint8_t, ADC2_AddressBase+0x06)
  ADC2 Schmitt trigger disable register.
- #define _ADC2_TDRL _SFR(uint8_t, ADC2_AddressBase+0x07)
  ADC2 Schmitt trigger disable register.
- #define _ADC2_CSR_RESET_VALUE ((uint8_t) 0x00)
  ADC2 control/status register reset value.
- #define _ADC2_CR1_RESET_VALUE ((uint8_t) 0x00)
  ADC2 Configuration Register 1 reset value.
- #define _ADC2_CR2_RESET_VALUE ((uint8_t) 0x00)

ADC2 Configuration Register 2 reset value.

- #define _ADC2_TDRL_RESET_VALUE ((uint8_t) 0x00)

  ADC2 Schmitt trigger disable register reset value.
- #define _ADC2_TDRH_RESET_VALUE ((uint8_t) 0x00)

  ADC2 Schmitt trigger disable register reset value.
- #define _ADC2_CH ((uint8_t) (0x0F << 0))

  ADC2 Channel selection bits [3:0] (in _ADC2_CSR)
- #define _ADC2_CH0 ((uint8_t) (0x01 << 0))

  ADC2 Channel selection bits [0] (in _ADC2_CSR)
- #define _ADC2_CH1 ((uint8_t) (0x01 << 1))

  ADC2 Channel selection bits [1] (in _ADC2_CSR)
- #define _ADC2_CH2 ((uint8_t) (0x01 << 2))

  ADC2 Channel selection bits [2] (in _ADC2_CSR)
- #define _ADC2_CH3 ((uint8_t) (0x01 << 3))

  ADC2 Channel selection bits [3] (in _ADC2_CSR)
- #define _ADC2_EOCIE ((uint8_t) (0x01 << 5))

  ADC2 Interrupt enable for EOC [0] (in _ADC2_CSR)
- #define _ADC2_EOC ((uint8_t) (0x01 << 7))

  ADC2 End of conversion [0] (in _ADC2_CSR)
- #define _ADC2_ADON ((uint8_t) (0x01 << 0))

  ADC2 Conversion on/off [0] (in _ADC2_CR1)
- #define _ADC2_CONT ((uint8_t) (0x01 << 1))

  ADC2 Continuous conversion [0] (in _ADC2_CR1)
- #define _ADC2_SPSEL ((uint8_t) (0x07 << 4)

  ADC2 clock prescaler selection [2:0] (in _ADC2_CR1)
- #define _ADC2_SPSEL0 ((uint8_t) (0x01 << 4)

  ADC2 clock prescaler selection [0] (in _ADC2_CR1)
- #define _ADC2_SPSEL1 ((uint8_t) (0x01 << 5)

  ADC2 clock prescaler selection [1] (in _ADC2_CR1)
- #define _ADC2_SPSEL2 ((uint8_t) (0x01 << 6)

  ADC2 clock prescaler selection [2] (in _ADC2_CR1)
- #define _ADC2_ALIGN ((uint8_t) (0x01 << 3))

  ADC2 Data alignment [0] (in _ADC2_CR2)
- #define _ADC2_EXTSEL ((uint8_t) (0x03 << 4)

  ADC2 External event selection [1:0] (in _ADC2_CR2)
- #define _ADC2_EXTSEL0 ((uint8_t) (0x01 << 4)

  ADC2 External event selection [0] (in _ADC2_CR2)
- #define _ADC2_EXTSEL1 ((uint8_t) (0x01 << 5)

  ADC2 External event selection [1] (in _ADC2_CR2)
- #define _ADC2_EXTTRIG ((uint8_t) (0x01 << 6))

  ADC2 External trigger enable [0] (in _ADC2_CR2)
- #define _CAN _SFR(CAN_t, CAN_AddressBase)

  CAN struct/bit access.
- #define _CAN_MCR _SFR(uint8_t, CAN_AddressBase+0x00)

  CAN master control register.
- #define _CAN_MSR _SFR(uint8_t, CAN_AddressBase+0x01)

  CAN master status register.
- #define _CAN_TSR _SFR(uint8_t, CAN_AddressBase+0x02)

  CAN transmit status register.
- #define _CAN_TPR _SFR(uint8_t, CAN_AddressBase+0x03)

  CAN transmit priority register.

- #define _CAN_RFR _SFR(uint8_t, CAN_AddressBase+0x04)

  CAN receive FIFO register.
- #define _CAN_IER _SFR(uint8_t, CAN_AddressBase+0x05)

  CAN interrupt enable register.
- #define _CAN_DGR _SFR(uint8_t, CAN_AddressBase+0x06)

  CAN diagnosis register.
- #define _CAN_PSR _SFR(uint8_t, CAN_AddressBase+0x07)

  CAN page selection for below paged registers.
- #define _CAN_MCSR _SFR(uint8_t, CAN_AddressBase+0x08+0x00)

  CAN message control/status register (page 0,1,5)
- #define _CAN_MDLCR _SFR(uint8_t, CAN_AddressBase+0x08+0x01)

  CAN mailbox data length control register (page 0,1,5,7)
- #define _CAN_MIDR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x02)

  CAN mailbox identifier register 1 (page 0,1,5,7)
- #define _CAN_MIDR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x03)

  CAN mailbox identifier register 2 (page 0,1,5,7)
- #define _CAN_MIDR3 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)

  CAN mailbox identifier register 3 (page 0,1,5,7)
- #define _CAN_MIDR4 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)

  CAN mailbox identifier register 4 (page 0,1,5,7)
- #define _CAN_MDAR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x06)

  CAN mailbox data register 1 (page 0,1,5,7) ∗/.
- #define _CAN_MDAR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x07)

  CAN mailbox data register 2 (page 0,1,5,7) ∗/.
- #define _CAN_MDAR3 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)

  CAN mailbox data register 3 (page 0,1,5,7) ∗/.
- #define _CAN_MDAR4 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)

  CAN mailbox data register 4 (page 0,1,5,7) ∗/.
- #define _CAN_MDAR5 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)

  CAN mailbox data register 5 (page 0,1,5,7) ∗/.
- #define _CAN_MDAR6 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)

  CAN mailbox data register 6 (page 0,1,5,7) ∗/.
- #define _CAN_MDAR7 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)

  CAN mailbox data register 7 (page 0,1,5,7) ∗/.
- #define _CAN_MDAR8 _SFR(uint8_t, CAN_AddressBase+0x08+0x0D)

  CAN mailbox data register 8 (page 0,1,5,7) ∗/.
- #define _CAN_MTSRL _SFR(uint8_t, CAN_AddressBase+0x08+0x0E)

  CAN mailbox time stamp register low byte (page 0,1,5,7) ∗/.
- #define _CAN_MTSRH _SFR(uint8_t, CAN_AddressBase+0x08+0x0F)

  CAN mailbox time stamp register high byte (page 0,1,5,7) ∗/.
- #define _CAN_F0R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x00)

  CAN acceptance filter 0/1 (page 2)
- #define _CAN_F0R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x01)

  CAN acceptance filter 0/2 (page 2)
- #define _CAN_F0R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x02)

  CAN acceptance filter 0/3 (page 2)
- #define _CAN_F0R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x03)

  CAN acceptance filter 0/4 (page 2)
- #define _CAN_F0R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)

  CAN acceptance filter 0/5 (page 2)
- #define _CAN_F0R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)

CAN acceptance filter 0/6 (page 2)

- #define _CAN_F0R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x06)

  CAN acceptance filter 0/7 (page 2)
- #define _CAN_F0R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x07)

  CAN acceptance filter 0/8 (page 2)
- #define _CAN_F1R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)

  CAN acceptance filter 1/1 (page 2)
- #define _CAN_F1R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)

  CAN acceptance filter 1/2 (page 2)
- #define _CAN_F1R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)

  CAN acceptance filter 1/3 (page 2)
- #define _CAN_F1R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)

  CAN acceptance filter 1/4 (page 2)
- #define _CAN_F1R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)

  CAN acceptance filter 1/5 (page 2)
- #define _CAN_F1R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x0D)

  CAN acceptance filter 1/6 (page 2)
- #define _CAN_F1R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x0E)

  CAN acceptance filter 1/7 (page 2)
- #define _CAN_F1R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x0F)

  CAN acceptance filter 1/8 (page 2)
- #define _CAN_F2R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x00)

  CAN acceptance filter 2/1 (page 3)
- #define _CAN_F2R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x01)

  CAN acceptance filter 2/2 (page 3)
- #define _CAN_F2R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x02)

  CAN acceptance filter 2/3 (page 3)
- #define _CAN_F2R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x03)

  CAN acceptance filter 2/4 (page 3)
- #define _CAN_F2R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)

  CAN acceptance filter 2/5 (page 3)
- #define _CAN_F2R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)

  CAN acceptance filter 2/6 (page 3)
- #define _CAN_F2R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x06)

  CAN acceptance filter 2/7 (page 3)
- #define _CAN_F2R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x07)

  CAN acceptance filter 2/8 (page 3)
- #define _CAN_F3R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)

  CAN acceptance filter 3/1 (page 3)
- #define _CAN_F3R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)

  CAN acceptance filter 3/2 (page 3)
- #define _CAN_F3R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)

  CAN acceptance filter 3/3 (page 3)
- #define _CAN_F3R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)

  CAN acceptance filter 3/4 (page 3)
- #define _CAN_F3R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)

  CAN acceptance filter 3/5 (page 3)
- #define _CAN_F3R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x0D)

  CAN acceptance filter 3/6 (page 3)
- #define _CAN_F3R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x0E)

  CAN acceptance filter 3/7 (page 3)

- #define _CAN_F3R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x0F)
  CAN acceptance filter 3/8 (page 3)
- #define _CAN_F4R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x00)
  CAN acceptance filter 4/1 (page 4)
- #define _CAN_F4R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x01)
  CAN acceptance filter 4/2 (page 4)
- #define _CAN_F4R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x02)
  CAN acceptance filter 4/3 (page 4)
- #define _CAN_F4R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x03)
  CAN acceptance filter 4/4 (page 4)
- #define _CAN_F4R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)
  CAN acceptance filter 4/5 (page 4)
- #define _CAN_F4R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)
  CAN acceptance filter 4/6 (page 4)
- #define _CAN_F4R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x06)
  CAN acceptance filter 4/7 (page 4)
- #define _CAN_F4R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x07)
  CAN acceptance filter 4/8 (page 4)
- #define _CAN_F5R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)
  CAN acceptance filter 5/1 (page 4)
- #define _CAN_F5R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)
  CAN acceptance filter 5/2 (page 4)
- #define _CAN_F5R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)
  CAN acceptance filter 5/3 (page 4)
- #define _CAN_F5R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)
  CAN acceptance filter 5/4 (page 4)
- #define _CAN_F5R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)
  CAN acceptance filter 5/5 (page 4)
- #define _CAN_F5R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x0D)
  CAN acceptance filter 5/6 (page 4)
- #define _CAN_F5R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x0E)
  CAN acceptance filter 5/7 (page 4)
- #define _CAN_F5R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x0F)
  CAN acceptance filter 5/8 (page 4)
- #define _CAN_ESR _SFR(uint8_t, CAN_AddressBase+0x08+0x00)
  CAN error status register (page 6)
- #define _CAN_EIER _SFR(uint8_t, CAN_AddressBase+0x08+0x01)
  CAN error interrupt enable register (page 6)
- #define _CAN_TECR _SFR(uint8_t, CAN_AddressBase+0x08+0x02)
  CAN transmit error counter register (page 6)
- #define _CAN_RECR _SFR(uint8_t, CAN_AddressBase+0x08+0x03)
  CAN receive error counter register (page 6)
- #define _CAN_BTR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)
  CAN bit timing register 1 (page 6)
- #define _CAN_BTR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)
  CAN bit timing register 2 (page 6)
- #define _CAN_FMR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)
  CAN filter mode register 1 (page 6)
- #define _CAN_FMR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)
  CAN filter mode register 2 (page 6)
- #define _CAN_FCR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)

CAN filter configuration register 1 (page 6)

- #define _CAN_FCR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)

  CAN filter configuration register 2 (page 6)
- #define _CAN_FCR3 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)

  CAN filter configuration register 3 (page 6)
- #define _CAN_MFMIR _SFR(uint8_t, CAN_AddressBase+0x08+0x00)

  CAN mailbox filter match index register (page 7)
- #define _CAN_MCR_RESET_VALUE ((uint8_t) 0x02)

  CAN master control register reset value.
- #define _CAN_MSR_RESET_VALUE ((uint8_t) 0x02)

  CAN master status register reset value.
- #define _CAN_TSR_RESET_VALUE ((uint8_t) 0x00)

  CAN transmit status register reset value.
- #define _CAN_TPR_RESET_VALUE ((uint8_t) 0x0C)

  CAN transmit priority register reset value.
- #define _CAN_RFR_RESET_VALUE ((uint8_t) 0x00)

  CAN receive FIFO register reset value.
- #define _CAN_IER_RESET_VALUE ((uint8_t) 0x00)

  CAN interrupt enable register reset value.
- #define _CAN_DGR_RESET_VALUE ((uint8_t) 0x0C)

  CAN diagnosis register reset value.
- #define _CAN_PSR_RESET_VALUE ((uint8_t) 0x00)

  CAN page selection reset value.
- #define _CAN_MCSR_RESET_VALUE ((uint8_t) 0x00)

  CAN message control/status register (page 0,1,5) reset value.
- #define _CAN_MDLCR_RESET_VALUE ((uint8_t) 0x00)

  CAN mailbox data length control register (page 0,1,5,7) reset value.
- #define _CAN_ESR_RESET_VALUE ((uint8_t) 0x00)

  CAN error status register (page 6) reset value.
- #define _CAN_EIER_RESET_VALUE ((uint8_t) 0x00)

  CAN error interrupt enable register (page 6) reset value.
- #define _CAN_TECR_RESET_VALUE ((uint8_t) 0x00)

  CAN transmit error counter register (page 6) reset value.
- #define _CAN_RECR_RESET_VALUE ((uint8_t) 0x00)

  CAN receive error counter register (page 6) reset value.
- #define _CAN_BTR1_RESET_VALUE ((uint8_t) 0x40)

  CAN bit timing register 1 (page 6) reset value.
- #define _CAN_BTR2_RESET_VALUE ((uint8_t) 0x23)

  CAN bit timing register 2 (page 6) reset value.
- #define _CAN_FMR1_RESET_VALUE ((uint8_t) 0x00)

  CAN filter mode register 1 (page 6) reset value.
- #define _CAN_FMR2_RESET_VALUE ((uint8_t) 0x00)

  CAN filter mode register 2 (page 6) reset value.
- #define _CAN_FCR_RESET_VALUE ((uint8_t) 0x00)

  CAN filter configuration register reset value.
- #define _CAN_MFMIR_RESET_VALUE ((uint8_t) 0x00)

  CAN mailbox filter match index register reset value.
- #define _CAN_INRQ ((uint8_t) (0x01 << 0))

  CAN Channel Initialization Request [0] (in _CAN_MCR)
- #define _CAN_SLEEP ((uint8_t) (0x01 << 1))

  CAN Channel Sleep Mode Request [0] (in _CAN_MCR)

- #define _CAN_TXFP ((uint8_t) (0x01 << 2))
  CAN Channel Transmit FIFO Priority [0] (in _CAN_MCR)
- #define _CAN_RFLM ((uint8_t) (0x01 << 3))
  CAN Channel Receive FIFO Locked Mode [0] (in _CAN_MCR)
- #define _CAN_NART ((uint8_t) (0x01 << 4))
  CAN Channel No Automatic Retransmission [0] (in _CAN_MCR)
- #define _CAN_AWUM ((uint8_t) (0x01 << 5))
  CAN Channel Automatic Wakeup Mode [0] (in _CAN_MCR)
- #define _CAN_ABOM ((uint8_t) (0x01 << 6))
  CAN Channel Automatic Bus-Off Management [0] (in _CAN_MCR)
- #define _CAN_TTCM ((uint8_t) (0x01 << 7))
  CAN Channel Time Triggered Communication Mode [0] (in _CAN_MCR)
- #define _CAN_INAK ((uint8_t) (0x01 << 0))
  CAN Initialization Acknowledge [0] (in _CAN_MSR)
- #define _CAN_SLAK ((uint8_t) (0x01 << 1))
  CAN Sleep Acknowledge [0] (in _CAN_MSR)
- #define _CAN_ERRI ((uint8_t) (0x01 << 2))
  CAN Error Interrupt [0] (in _CAN_MSR)
- #define _CAN_WKUI ((uint8_t) (0x01 << 3))
  CAN Wakeup Interrupt [0] (in _CAN_MSR)
- #define _CAN_TX ((uint8_t) (0x01 << 4))
  CAN Transmit [0] (in _CAN_MSR)
- #define _CAN_RX ((uint8_t) (0x01 << 5))
  CAN Receive [0] (in _CAN_MSR)
- #define _CAN_RQCP0 ((uint8_t) (0x01 << 0))
  CAN Request Completed for Mailbox 0 [0] (in _CAN_TSR)
- #define _CAN_RQCP1 ((uint8_t) (0x01 << 1))
  CAN Request Completed for Mailbox 1 [0] (in _CAN_TSR)
- #define _CAN_RQCP2 ((uint8_t) (0x01 << 2))
  CAN Request Completed for Mailbox 2 [0] (in _CAN_TSR)
- #define _CAN_TXOK0 ((uint8_t) (0x01 << 4))
  CAN Transmission ok for Mailbox 0 [0] (in _CAN_TSR)
- #define _CAN_TXOK1 ((uint8_t) (0x01 << 5))
  CAN Transmission ok for Mailbox 1 [0] (in _CAN_TSR)
- #define _CAN_TXOK2 ((uint8_t) (0x01 << 6))
  CAN Transmission ok for Mailbox 2 [0] (in _CAN_TSR)
- #define _CAN_CODE ((uint8_t) (0x03 << 0))
  CAN Mailbox Code [1:0] (in _CAN_TPR)
- #define _CAN_CODE0 ((uint8_t) (0x01 << 0))
  CAN Mailbox Code [0] (in _CAN_TPR)
- #define _CAN_CODE1 ((uint8_t) (0x01 << 1))
  CAN Mailbox Code [1] (in _CAN_TPR)
- #define _CAN_TME0 ((uint8_t) (0x01 << 2))
  CAN Transmit Mailbox 0 Empty [0] (in _CAN_TPR)
- #define _CAN_TME1 ((uint8_t) (0x01 << 3))
  CAN Transmit Mailbox 1 Empty [0] (in _CAN_TPR)
- #define _CAN_TME2 ((uint8_t) (0x01 << 4))
  CAN Transmit Mailbox 2 Empty [0] (in _CAN_TPR)
- #define _CAN_LOW0 ((uint8_t) (0x01 << 5))
  CAN Lowest Priority Flag for Mailbox 0 [0] (in _CAN_TPR)
- #define _CAN_LOW1 ((uint8_t) (0x01 << 6))

CAN Lowest Priority Flag for Mailbox 1 [0] (in _CAN_TPR)

- #define _CAN_LOW2 ((uint8_t) (0x01 << 7))

  CAN Lowest Priority Flag for Mailbox 2 [0] (in _CAN_TPR)
- #define _CAN_FMP ((uint8_t) (0x03 << 0))

  CAN FIFO Message Pending [1:0] (in _CAN_RFR)
- #define _CAN_FMP0 ((uint8_t) (0x01 << 0))

  CAN FIFO Message Pending [0] (in _CAN_RFR)
- #define _CAN_FMP1 ((uint8_t) (0x01 << 1))

  CAN FIFO Message Pending [1] (in _CAN_RFR)
- #define _CAN_FULL ((uint8_t) (0x01 << 3))

  CAN FIFO Full [0] (in _CAN_RFR)
- #define _CAN_FOVR ((uint8_t) (0x01 << 4))

  CAN FIFO Overrun [0] (in _CAN_RFR)
- #define _CAN_RFOM ((uint8_t) (0x01 << 5))

  CAN Release FIFO Output Mailbox [0] (in _CAN_RFR)
- #define _CAN_TMEIE ((uint8_t) (0x01 << 0))

  CAN Transmit Mailbox Empty Interrupt Enable [0] (in _CAN_IER)
- #define _CAN_FMPIE ((uint8_t) (0x01 << 1))

  CAN FIFO Message Pending Interrupt Enable [0] (in _CAN_IER)
- #define _CAN_FFIE ((uint8_t) (0x01 << 2))

  CAN FIFO Full Interrupt Enable [0] (in _CAN_IER)
- #define _CAN_FOVIE ((uint8_t) (0x01 << 3))

  CAN FIFO Overrun Interrupt Enable [0] (in _CAN_IER)
- #define _CAN_WKUIE ((uint8_t) (0x01 << 7))

  CAN Wakeup Interrupt Enable [0] (in _CAN_IER)
- #define _CAN_LBKM ((uint8_t) (0x01 << 0))

  CAN Loop back mode [0] (in _CAN_DGR)
- #define _CAN_SILM ((uint8_t) (0x01 << 1))

  CAN Silent mode [0] (in _CAN_DGR)
- #define _CAN_SAMP ((uint8_t) (0x01 << 2))

  CAN Last sample point [0] (in _CAN_DGR)
- #define _CAN_RXS ((uint8_t) (0x01 << 3))

  CAN Rx Signal (=pin status) [0] (in _CAN_DGR)
- #define _CAN_TXM2E ((uint8_t) (0x01 << 4))

  CAN TX Mailbox 2 enable [0] (in _CAN_DGR)
- #define _CAN_PS ((uint8_t) (0x07 << 0))

  CAN Page select [2:0] (in _CAN_PSR)
- #define _CAN_PS0 ((uint8_t) (0x01 << 0))

  CAN Page select [0] (in _CAN_PSR)
- #define _CAN_PS1 ((uint8_t) (0x01 << 1))

  CAN Page select [1] (in _CAN_PSR)
- #define _CAN_PS2 ((uint8_t) (0x01 << 2))

  CAN Page select [2] (in _CAN_PSR)
- #define _CAN_TXRQ ((uint8_t) (0x01 << 0))

  CAN Transmission mailbox request [0] (in _CAN_MCSR, page 0,1,5)
- #define _CAN_ABRQ ((uint8_t) (0x01 << 1))

  CAN Abort request for mailbox [0] (in _CAN_MCSR, page 0,1,5)
- #define _CAN_RQCP ((uint8_t) (0x01 << 2))

  CAN Request completed [0] (in _CAN_MCSR, page 0,1,5)
- #define _CAN_TXOK ((uint8_t) (0x01 << 3))

  CAN Transmission OK [0] (in _CAN_MCSR, page 0,1,5)

- #define _CAN_ALST ((uint8_t) (0x01 << 4))
  CAN Arbitration lost [0] (in _CAN_MCSR, page 0,1,5)
- #define _CAN_TERR ((uint8_t) (0x01 << 5))
  CAN Transmission error [0] (in _CAN_MCSR, page 0,1,5)
- #define _CAN_DLC ((uint8_t) (0x0F << 0))
  CAN Data length code [3:0] (in _CAN_MDLCR, page 0,1,5,7)
- #define _CAN_DLC0 ((uint8_t) (0x01 << 0))
  CAN Data length code [0] (in _CAN_MDLCR, page 0,1,5,7)
- #define _CAN_DLC1 ((uint8_t) (0x01 << 1))
  CAN Data length code [1] (in _CAN_MDLCR, page 0,1,5,7)
- #define _CAN_DLC2 ((uint8_t) (0x01 << 2))
  CAN Data length code [2] (in _CAN_MDLCR, page 0,1,5,7)
- #define _CAN_DLC3 ((uint8_t) (0x01 << 3))
  CAN Data length code [3] (in _CAN_MDLCR, page 0,1,5,7)
- #define _CAN_TGT ((uint8_t) (0x01 << 7))
  CAN Transmit global time [0] (in _CAN_MDLCR, page 0,1,5,7)
- #define _CAN_RTR ((uint8_t) (0x01 << 5))
  CAN Remote transmission request [0] (in _CAN_MIDR1, page 0,1,5)
- #define _CAN_IDE ((uint8_t) (0x01 << 6))
  CAN Extended identifier [0] (in _CAN_MIDR1, page 0,1,5)
- #define _CAN_EWGF ((uint8_t) (0x01 << 0))
  CAN Error warning flag [0] (in _CAN_ESR, page 6)
- #define _CAN_EPVF ((uint8_t) (0x01 << 1))
  CAN Error passive flag [0] (in _CAN_ESR, page 6)
- #define _CAN_BOFF ((uint8_t) (0x01 << 2))
  CAN Bus off flag [0] (in _CAN_ESR, page 6)
- #define _CAN_LEC ((uint8_t) (0x07 << 4))
  CAN Last error code [2:0] (in _CAN_ESR, page 6)
- #define _CAN_LEC0 ((uint8_t) (0x01 << 4))
  CAN Last error code [0] (in _CAN_ESR, page 6)
- #define _CAN_LEC1 ((uint8_t) (0x01 << 5))
  CAN Last error code [1] (in _CAN_ESR, page 6)
- #define _CAN_LEC2 ((uint8_t) (0x01 << 6))
  CAN Last error code [3] (in _CAN_ESR, page 6)
- #define _CAN_EWGIE ((uint8_t) (0x01 << 0))
  CAN Error warning interrupt enable [0] (in _CAN_EIER, page 6)
- #define _CAN_EPVIE ((uint8_t) (0x01 << 1))
  CAN Error passive interrupt enable [0] (in _CAN_EIER, page 6)
- #define _CAN_BOFIE ((uint8_t) (0x01 << 2))
  CAN Bus-Off interrupt enable [0] (in _CAN_EIER, page 6)
- #define _CAN_LECIE ((uint8_t) (0x01 << 4))
  CAN Last error code interrupt enable [0] (in _CAN_EIER, page 6)
- #define _CAN_ERRIE ((uint8_t) (0x01 << 6))
  CAN Error interrupt enable [0] (in _CAN_EIER, page 6)
- #define _CAN_BRP ((uint8_t) (0x3F << 0))
  CAN Baud rate prescaler [5:0] (in _CAN_BTR1, page 6)
- #define _CAN_BRP0 ((uint8_t) (0x01 << 0))
  CAN Baud rate prescaler [0] (in _CAN_BTR1, page 6)
- #define _CAN_BRP1 ((uint8_t) (0x01 << 1))
  CAN Baud rate prescaler [1] (in _CAN_BTR1, page 6)
- #define _CAN_BRP2 ((uint8_t) (0x01 << 2))

CAN Baud rate prescaler [2] (in _CAN_BTR1, page 6)

- #define _CAN_BRP3 ((uint8_t) (0x01 << 3))

  CAN Baud rate prescaler [3] (in _CAN_BTR1, page 6)
- #define _CAN_BRP4 ((uint8_t) (0x01 << 4))

  CAN Baud rate prescaler [4] (in _CAN_BTR1, page 6)
- #define _CAN_BRP5 ((uint8_t) (0x01 << 5))

  CAN Baud rate prescaler [5] (in _CAN_BTR1, page 6)
- #define _CAN_SJW ((uint8_t) (0x03 << 6))

  CAN Resynchronization jump width [1:0] (in _CAN_EIER, page 6)
- #define _CAN_SJW0 ((uint8_t) (0x01 << 6))

  CAN Resynchronization jump width [0] (in _CAN_EIER, page 6)
- #define _CAN_SJW1 ((uint8_t) (0x01 << 7))

  CAN Resynchronization jump width [1] (in _CAN_EIER, page 6)
- #define _CAN_BS1 ((uint8_t) (0x0F << 0))

  CAN Bit segment 1 [3:0] (in _CAN_BTR2, page 6)
- #define _CAN_BS10 ((uint8_t) (0x01 << 0))

  CAN Bit segment 1 [0] (in _CAN_BTR2, page 6)
- #define _CAN_BS11 ((uint8_t) (0x01 << 1))

  CAN Bit segment 1 [1] (in _CAN_BTR2, page 6)
- #define _CAN_BS12 ((uint8_t) (0x01 << 2))

  CAN Bit segment 1 [2] (in _CAN_BTR2, page 6)
- #define _CAN_BS13 ((uint8_t) (0x01 << 3))

  CAN Bit segment 1 [3] (in _CAN_BTR2, page 6)
- #define _CAN_BS2 ((uint8_t) (0x07 << 4))

  CAN Bit segment 2 [2:0] (in _CAN_BTR2, page 6)
- #define _CAN_BS20 ((uint8_t) (0x01 << 4))

  CAN Bit segment 2 [0] (in _CAN_BTR2, page 6)
- #define _CAN_BS21 ((uint8_t) (0x01 << 5))

  CAN Bit segment 2 [1] (in _CAN_BTR2, page 6)
- #define _CAN_BS22 ((uint8_t) (0x01 << 6))

  CAN Bit segment 2 [2] (in _CAN_BTR2, page 6)
- #define _CAN_FML0 ((uint8_t) (0x01 << 0))

  CAN Filter 0 mode low [0] (in _CAN_FMR1, page 6)
- #define _CAN_FMH0 ((uint8_t) (0x01 << 1))

  CAN Filter 0 mode high [0] (in _CAN_FMR1, page 6)
- #define _CAN_FML1 ((uint8_t) (0x01 << 2))

  CAN Filter 1 mode low [0] (in _CAN_FMR1, page 6)
- #define _CAN_FMH1 ((uint8_t) (0x01 << 3))

  CAN Filter 1 mode high [0] (in _CAN_FMR1, page 6)
- #define _CAN_FML2 ((uint8_t) (0x01 << 4))

  CAN Filter 2 mode low [0] (in _CAN_FMR1, page 6)
- #define _CAN_FMH2 ((uint8_t) (0x01 << 5))

  CAN Filter 2 mode high [0] (in _CAN_FMR1, page 6)
- #define _CAN_FML3 ((uint8_t) (0x01 << 6))

  CAN Filter 3 mode low [0] (in _CAN_FMR1, page 6)
- #define _CAN_FMH3 ((uint8_t) (0x01 << 7))

  CAN Filter 3 mode high [0] (in _CAN_FMR1, page 6)
- #define _CAN_FML4 ((uint8_t) (0x01 << 0))

  CAN Filter 4 mode low [0] (in _CAN_FMR2, page 6)
- #define _CAN_FMH4 ((uint8_t) (0x01 << 1))

  CAN Filter 4 mode high [0] (in _CAN_FMR2, page 6)

- #define _CAN_FML5 ((uint8_t) (0x01 << 2))
  CAN Filter 5 mode low [0] (in _CAN_FMR2, page 6)
- #define _CAN_FMH5 ((uint8_t) (0x01 << 3))
  CAN Filter 5 mode high [0] (in _CAN_FMR2, page 6)
- #define _CAN_FACT0 ((uint8_t) (0x01 << 0))
  CAN Filter 0 active [0] (in _CAN_FCR1, page 6)
- #define _CAN_FSC0 ((uint8_t) (0x03 << 1))
  CAN Filter 0 scale configuration [1:0] (in _CAN_FCR1, page 6)
- #define _CAN_FSC00 ((uint8_t) (0x01 << 1))
  CAN Filter 0 scale configuration [0] (in _CAN_FCR1, page 6)
- #define _CAN_FSC01 ((uint8_t) (0x01 << 2))
  CAN Filter 0 scale configuration [1] (in _CAN_FCR1, page 6)
- #define _CAN_FACT1 ((uint8_t) (0x01 << 4))
  CAN Filter 1 active [0] (in _CAN_FCR1, page 6)
- #define _CAN_FSC1 ((uint8_t) (0x03 << 5))
  CAN Filter 1 scale configuration [1:0] (in _CAN_FCR1, page 6)
- #define _CAN_FSC10 ((uint8_t) (0x01 << 5))
  CAN Filter 1 scale configuration [0] (in _CAN_FCR1, page 6)
- #define _CAN_FSC11 ((uint8_t) (0x01 << 6))
  CAN Filter 1 scale configuration [1] (in _CAN_FCR1, page 6)
- #define _CAN_FACT2 ((uint8_t) (0x01 << 0))
  CAN Filter 2 active [0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC2 ((uint8_t) (0x03 << 1))
  CAN Filter 2 scale configuration [1:0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC20 ((uint8_t) (0x01 << 1))
  CAN Filter 2 scale configuration [0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC21 ((uint8_t) (0x01 << 2))
  CAN Filter 2 scale configuration [1] (in _CAN_FCR2, page 6)
- #define _CAN_FACT3 ((uint8_t) (0x01 << 4))
  CAN Filter 3 active [0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC3 ((uint8_t) (0x03 << 5))
  CAN Filter 3 scale configuration [1:0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC30 ((uint8_t) (0x01 << 5))
  CAN Filter 3 scale configuration [0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC31 ((uint8_t) (0x01 << 6))
  CAN Filter 3 scale configuration [1] (in _CAN_FCR2, page 6)
- #define _CAN_FACT4 ((uint8_t) (0x01 << 0))
  CAN Filter 4 active [0] (in _CAN_FCR3, page 6)
- #define _CAN_FSC4 ((uint8_t) (0x03 << 1))
  CAN Filter 4 scale configuration [1:0] (in _CAN_FCR3, page 6)
- #define _CAN_FSC40 ((uint8_t) (0x01 << 1))
  CAN Filter 4 scale configuration [0] (in _CAN_FCR3, page 6)
- #define _CAN_FSC41 ((uint8_t) (0x01 << 2))
  CAN Filter 4 scale configuration [1] (in _CAN_FCR3, page 6)
- #define _CAN_FACT5 ((uint8_t) (0x01 << 4))
  CAN Filter 5 active [0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC5 ((uint8_t) (0x03 << 5))
  CAN Filter 5 scale configuration [1:0] (in _CAN_FCR3, page 6)
- #define _CAN_FSC50 ((uint8_t) (0x01 << 5))
  CAN Filter 5 scale configuration [0] (in _CAN_FCR3, page 6)
- #define _CAN_FSC51 ((uint8_t) (0x01 << 6))

CAN Filter 5 scale configuration [1] (in _CAN_FCR3, page 6)

- #define _CFG _SFR(CFG_t, CFG_AddressBase)

  CFG struct/bit access.
- #define _CFG_GCR _SFR(uint8_t, CFG_AddressBase+0x00)

  Global configuration register (CFG_GCR)
- #define _CFG_GCR_RESET_VALUE ((uint8_t)0x00)
- #define _CFG_SWD ((uint8_t) (0x01 << 0))

  SWIM disable [0].
- #define _CFG_AL ((uint8_t) (0x01 << 1))

  Activation level [0].
- #define _ITC _SFR(ITC_t, ITC_AddressBase)

  ITC struct/bit access.
- #define _ITC_SPR1 _SFR(uint8_t, ITC_AddressBase+0x00)

  Interrupt priority register 1/8.
- #define _ITC_SPR2 _SFR(uint8_t, ITC_AddressBase+0x01)

  Interrupt priority register 2/8.
- #define _ITC_SPR3 _SFR(uint8_t, ITC_AddressBase+0x02)

  Interrupt priority register 3/8.
- #define _ITC_SPR4 _SFR(uint8_t, ITC_AddressBase+0x03)

  Interrupt priority register 4/8.
- #define _ITC_SPR5 _SFR(uint8_t, ITC_AddressBase+0x04)

  Interrupt priority register 5/8.
- #define _ITC_SPR6 _SFR(uint8_t, ITC_AddressBase+0x05)

  Interrupt priority register 6/8.
- #define _ITC_SPR7 _SFR(uint8_t, ITC_AddressBase+0x06)

  Interrupt priority register 7/8.
- #define _ITC_SPR8 _SFR(uint8_t, ITC_AddressBase+0x07)

  Interrupt priority register 8/8.
- #define _ITC_SPR1_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 1/8 reset value.
- #define _ITC_SPR2_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 2/8 reset value.
- #define _ITC_SPR3_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 3/8 reset value.
- #define _ITC_SPR4_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 4/8 reset value.
- #define _ITC_SPR5_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 5/8 reset value.
- #define _ITC_SPR6_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 6/8 reset value.
- #define _ITC_SPR7_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 7/8 reset value.
- #define _ITC_SPR8_RESET_VALUE ((uint8_t) 0x0F)

  Interrupt priority register 8/8 reset value.
- #define _ITC_VECT1SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 1 [1:0] (in _ITC_SPR1)
- #define _ITC_VECT1SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 1 [0] (in _ITC_SPR1)
- #define _ITC_VECT1SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 1 [1] (in _ITC_SPR1)
- #define _ITC_VECT2SPR ((uint8_t) (0x03 << 4))

ITC interrupt priority vector 2 [1:0] (in _ITC_SPR1)

- #define _ITC_VECT2SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 2 [0] (in _ITC_SPR1)
- #define _ITC_VECT2SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 2 [1] (in _ITC_SPR1)
- #define _ITC_VECT3SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 3 [1:0] (in _ITC_SPR1)
- #define _ITC_VECT3SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 3 [0] (in _ITC_SPR1)
- #define _ITC_VECT3SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 3 [1] (in _ITC_SPR1)
- #define _ITC_VECT4SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 4 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT4SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 4 [0] (in _ITC_SPR2)
- #define _ITC_VECT4SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 4 [1] (in _ITC_SPR2)
- #define _ITC_VECT5SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 5 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT5SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 5 [0] (in _ITC_SPR2)
- #define _ITC_VECT5SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 5 [1] (in _ITC_SPR2)
- #define _ITC_VECT6SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 6 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT6SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 6 [0] (in _ITC_SPR2)
- #define _ITC_VECT6SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 6 [1] (in _ITC_SPR2)
- #define _ITC_VECT7SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 7 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT7SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 7 [0] (in _ITC_SPR2)
- #define _ITC_VECT7SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 7 [1] (in _ITC_SPR2)
- #define _ITC_VECT8SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 8 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT8SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 8 [0] (in _ITC_SPR3)
- #define _ITC_VECT8SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 8 [1] (in _ITC_SPR3)
- #define _ITC_VECT9SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 9 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT9SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 9 [0] (in _ITC_SPR3)
- #define _ITC_VECT9SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 9 [1] (in _ITC_SPR3)
- #define _ITC_VECT10SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 10 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT10SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 10 [0] (in _ITC_SPR3)

- #define _ITC_VECT10SPR1 ((uint8_t) (0x01 << 5))
  ITC interrupt priority vector 10 [1] (in _ITC_SPR3)
- #define _ITC_VECT11SPR ((uint8_t) (0x03 << 6))
  ITC interrupt priority vector 11 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT11SPR0 ((uint8_t) (0x01 << 6))
  ITC interrupt priority vector 11 [0] (in _ITC_SPR3)
- #define _ITC_VECT11SPR1 ((uint8_t) (0x01 << 7))
  ITC interrupt priority vector 11 [1] (in _ITC_SPR3)
- #define _ITC_VECT12SPR ((uint8_t) (0x03 << 0))
  ITC interrupt priority vector 12 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT12SPR0 ((uint8_t) (0x01 << 0))
  ITC interrupt priority vector 12 [0] (in _ITC_SPR4)
- #define _ITC_VECT12SPR1 ((uint8_t) (0x01 << 1))
  ITC interrupt priority vector 12 [1] (in _ITC_SPR4)
- #define _ITC_VECT13SPR ((uint8_t) (0x03 << 2))
  ITC interrupt priority vector 13 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT13SPR0 ((uint8_t) (0x01 << 2))
  ITC interrupt priority vector 13 [0] (in _ITC_SPR4)
- #define _ITC_VECT13SPR1 ((uint8_t) (0x01 << 3))
  ITC interrupt priority vector 13 [1] (in _ITC_SPR4)
- #define _ITC_VECT14SPR ((uint8_t) (0x03 << 4))
  ITC interrupt priority vector 14 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT14SPR0 ((uint8_t) (0x01 << 4))
  ITC interrupt priority vector 14 [0] (in _ITC_SPR4)
- #define _ITC_VECT14SPR1 ((uint8_t) (0x01 << 5))
  ITC interrupt priority vector 14 [1] (in _ITC_SPR4)
- #define _ITC_VECT15SPR ((uint8_t) (0x03 << 6))
  ITC interrupt priority vector 15 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT15SPR0 ((uint8_t) (0x01 << 6))
  ITC interrupt priority vector 15 [0] (in _ITC_SPR4)
- #define _ITC_VECT15SPR1 ((uint8_t) (0x01 << 7))
  ITC interrupt priority vector 15 [1] (in _ITC_SPR4)
- #define _ITC_VECT16SPR ((uint8_t) (0x03 << 0))
  ITC interrupt priority vector 16 [1:0] (in _ITC_SPR5)
- #define _ITC_VECT16SPR0 ((uint8_t) (0x01 << 0))
  ITC interrupt priority vector 16 [0] (in _ITC_SPR5)
- #define _ITC_VECT16SPR1 ((uint8_t) (0x01 << 1))
  ITC interrupt priority vector 16 [1] (in _ITC_SPR5)
- #define _ITC_VECT17SPR ((uint8_t) (0x03 << 2))
  ITC interrupt priority vector 17 [1:0] (in _ITC_SPR5)
- #define _ITC_VECT17SPR0 ((uint8_t) (0x01 << 2))
  ITC interrupt priority vector 17 [0] (in _ITC_SPR5)
- #define _ITC_VECT17SPR1 ((uint8_t) (0x01 << 3))
  ITC interrupt priority vector 17 [1] (in _ITC_SPR5)
- #define _ITC_VECT18SPR ((uint8_t) (0x03 << 4))
  ITC interrupt priority vector 18 [1:0] (in _ITC_SPR5)
- #define _ITC_VECT18SPR0 ((uint8_t) (0x01 << 4))
  ITC interrupt priority vector 18 [0] (in _ITC_SPR5)
- #define _ITC_VECT18SPR1 ((uint8_t) (0x01 << 5))
  ITC interrupt priority vector 18 [1] (in _ITC_SPR5)
- #define _ITC_VECT19SPR ((uint8_t) (0x03 << 6))

ITC interrupt priority vector 19 [1:0] (in _ITC_SPR5)

- #define _ITC_VECT19SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 19 [0] (in _ITC_SPR5)
- #define _ITC_VECT19SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 19 [1] (in _ITC_SPR5)
- #define _ITC_VECT20SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 20 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT20SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 20 [0] (in _ITC_SPR6)
- #define _ITC_VECT20SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 20 [1] (in _ITC_SPR6)
- #define _ITC_VECT21SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 21 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT21SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 21 [0] (in _ITC_SPR6)
- #define _ITC_VECT21SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 21 [1] (in _ITC_SPR6)
- #define _ITC_VECT22SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 22 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT22SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 22 [0] (in _ITC_SPR6)
- #define _ITC_VECT22SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 22 [1] (in _ITC_SPR6)
- #define _ITC_VECT23SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 23 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT23SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 23 [0] (in _ITC_SPR6)
- #define _ITC_VECT23SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 23 [1] (in _ITC_SPR6)
- #define _ITC_VECT24SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 24 [1:0] (in _ITC_SPR7)
- #define _ITC_VECT24SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 24 [0] (in _ITC_SPR7)
- #define _ITC_VECT24SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 24 [1] (in _ITC_SPR7)
- #define _ITC_VECT25SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 25 [1:0] (in _ITC_SPR7)
- #define _ITC_VECT25SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 25 [0] (in _ITC_SPR7)
- #define _ITC_VECT25SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 25 [1] (in _ITC_SPR7)
- #define _ITC_VECT26SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 26 [1:0] (in _ITC_SPR7)
- #define _ITC_VECT26SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 26 [0] (in _ITC_SPR7)
- #define _ITC_VECT26SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 26 [1] (in _ITC_SPR7)
- #define _ITC_VECT27SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 27 [1:0] (in _ITC_SPR7)
- #define _ITC_VECT27SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 27 [0] (in _ITC_SPR7)

- #define _ITC_VECT27SPR1 ((uint8_t) (0x01 << 7))
  ITC interrupt priority vector 27 [1] (in _ITC_SPR7)
- #define _ITC_VECT28SPR ((uint8_t) (0x03 << 0))
  ITC interrupt priority vector 28 [1:0] (in _ITC_SPR8)
- #define _ITC_VECT28SPR0 ((uint8_t) (0x01 << 0))
  ITC interrupt priority vector 28 [0] (in _ITC_SPR8)
- #define _ITC_VECT28SPR1 ((uint8_t) (0x01 << 1))
  ITC interrupt priority vector 28 [1] (in _ITC_SPR8)
- #define _ITC_VECT29SPR ((uint8_t) (0x03 << 2))
  ITC interrupt priority vector 29 [1:0] (in _ITC_SPR8)
- #define _ITC_VECT29SPR0 ((uint8_t) (0x01 << 2))
  ITC interrupt priority vector 29 [0] (in _ITC_SPR8)
- #define _ITC_VECT29SPR1 ((uint8_t) (0x01 << 3))
  ITC interrupt priority vector 29 [1] (in _ITC_SPR8)
- #define STM8S001J3
- #define STM8S001
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 128
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8S003F3
- #define STM8S003
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 128
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A

- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8S003K3
- #define STM8S003
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 128
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8S005C6
- #define STM8S005
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048

- #define STM8_EEPROM_SIZE 128
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8S005K6
- #define STM8S005
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 128
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320

- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8S007C8
- #define STM8S007
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 128
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define STM8S103F2
- #define STM8S103
- #define STM8_PFLASH_SIZE 4096
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3

- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4865
- #define STM8S103F3
- #define STM8S103
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4865
- #define STM8S103K3
- #define STM8S103
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005

- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4865
- #define STM8S105C4
- #define STM8S105
- #define STM8_PFLASH_SIZE 16384
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

- #define UID_AddressBase 0x48CD
- #define STM8S105C6
- #define STM8S105
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S105K4
- #define STM8S105
- #define STM8_PFLASH_SIZE 16384
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3

- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S105K6
- #define STM8S105
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S105S4
- #define STM8S105
- #define STM8_PFLASH_SIZE 16384
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F

- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S105S6
- #define STM8S105
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

- #define UID_AddressBase 0x48CD
- #define STM8S207C6
- #define STM8S207
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S207C8
- #define STM8S207
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3

- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S207CB
- #define STM8S207
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S207K6

- #define STM8S207
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S207K8
- #define STM8S207
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1

- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S207M8
- #define STM8S207
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S207MB
- #define STM8S207
- #define STM8_PFLASH_SIZE 131072

- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S207R6
- #define STM8S207
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0

- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S207R8
- #define STM8S207
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S207RB
- #define STM8S207
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048

- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S207S6
- #define STM8S207
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200

- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S207S8
- #define STM8S207
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S207SB
- #define STM8S207
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000

- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S208C6
- #define STM8S208
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230

- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S208C8
- #define STM8S208
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S208CB
- #define STM8S208
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000

- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S208MB
- #define STM8S208
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210

- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S208R6
- #define STM8S208
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S208R8
- #define STM8S208
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800

- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S208RB
- #define STM8S208
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200

- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S208S6
- #define STM8S208
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S208S8
- #define STM8S208
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536

- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S208SB
- #define STM8S208
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3

- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD
- #define STM8S903F3
- #define STM8S903
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4865
- #define STM8S903K3
- #define STM8S903
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F

- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4865

5.1.1 Detailed Description

5.1.2 Macro Definition Documentation

5.1.2.1 __ADC1_VECTOR__

```
#define __ADC1_VECTOR__ 22
```

irq22 - ADC1 end of conversion (shared with __ADC2_VECTOR__)

Definition at line 301 of file STM8AF_STM8S.h.

5.1.2.2 __AWU_VECTOR__

```
#define __AWU_VECTOR__ 1
```

irq1 - Auto Wake Up from Halt interrupt (AWU)

Definition at line 243 of file STM8AF_STM8S.h.

5.1.2.3 __CAN_RX_VECTOR__

```
#define __CAN_RX_VECTOR__ 8
```

irq8 - CAN receive interrupt (shared with __PORTF_VECTOR__)

Definition at line 251 of file STM8AF_STM8S.h.

5.1.2.4 __CAN_TX_VECTOR__

```
#define __CAN_TX_VECTOR__ 9
```

irq9 - CAN transmit interrupt

Definition at line 257 of file STM8AF_STM8S.h.

5.1.2.5 __CLK_VECTOR__

```
#define __CLK_VECTOR__ 2
```

irq2 - Clock Controller interrupt

Definition at line 244 of file STM8AF_STM8S.h.

5.1.2.6 __FLASH_VECTOR__

```
#define __FLASH_VECTOR__ 24
```

irq24 - flash interrupt

Definition at line 310 of file STM8AF_STM8S.h.

5.1.2.7 __I2C_VECTOR__

```
#define __I2C_VECTOR__ 19
```

irq19 - I2C interrupt

Definition at line 285 of file STM8AF_STM8S.h.

5.1.2.8 __PORTA_VECTOR__

```
#define __PORTA_VECTOR__ 3
```

irq3 - External interrupt 0 (GPIOA)

Definition at line 245 of file STM8AF_STM8S.h.

5.1.2.9 __PORTB_VECTOR__

```
#define __PORTB_VECTOR__ 4
```

irq4 - External interrupt 1 (GPIOB)

Definition at line 246 of file STM8AF_STM8S.h.

5.1.2.10 __PORTC_VECTOR__

```
#define __PORTC_VECTOR__ 5
```

irq5 - External interrupt 2 (GPIOC)

Definition at line 247 of file STM8AF_STM8S.h.

5.1.2.11 __PORTD_VECTOR__

```
#define __PORTD_VECTOR__ 6
```

irq6 - External interrupt 3 (GPIOD)

Definition at line 248 of file STM8AF_STM8S.h.

5.1.2.12 __PORTE_VECTOR__

```
#define __PORTE_VECTOR__ 7
```

irq7 - External interrupt 4 (GPIOE)

Definition at line 249 of file STM8AF_STM8S.h.

5.1.2.13 __PORTF_VECTOR__

```
#define __PORTF_VECTOR__ 8
```

irq8 - External interrupt 5 (GPIOF, shared with __CAN_RX_VECTOR__)

Definition at line 254 of file STM8AF_STM8S.h.

5.1.2.14 __SPI_VECTOR__

```
#define __SPI_VECTOR__ 10
```

irq10 - SPI End of transfer interrupt

Definition at line 259 of file STM8AF_STM8S.h.

5.1.2.15 __TIM1_CAPCOM_VECTOR__

```
#define __TIM1_CAPCOM_VECTOR__ 12
```

irq12 - TIM1 Capture/Compare interrupt

Definition at line 261 of file STM8AF_STM8S.h.

5.1.2.16 __TIM1_UPD_OVF_VECTOR__

```
#define __TIM1_UPD_OVF_VECTOR__ 11
```

irq11 - TIM1 Update/Overflow/Trigger/Break interrupt

Definition at line 260 of file STM8AF_STM8S.h.

5.1.2.17 __TIM2_CAPCOM_VECTOR__

```
#define __TIM2_CAPCOM_VECTOR__ 14
```

irq14 - TIM2 Capture/Compare interrupt (shared with __TIM5_CAPCOM_VECTOR__)

Definition at line 269 of file STM8AF_STM8S.h.

5.1.2.18 __TIM2_UPD_OVF_VECTOR__

```
#define __TIM2_UPD_OVF_VECTOR__ 13
```

irq13 - TIM2 Update/overflow interrupt (shared with __TIM5_UPD_OVF_VECTOR__)

Definition at line 263 of file STM8AF_STM8S.h.

5.1.2.19 __TIM3_CAPCOM_VECTOR__

```
#define __TIM3_CAPCOM_VECTOR__ 16
```

irq16 - TIM3 Capture/Compare interrupt

Definition at line 277 of file STM8AF_STM8S.h.

5.1.2.20 __TIM3_UPD_OVF_VECTOR__

```
#define __TIM3_UPD_OVF_VECTOR__ 15
```

irq15 - TIM3 Update/overflow interrupt

Definition at line 274 of file STM8AF_STM8S.h.

5.1.2.21 __TIM4_UPD_OVF_VECTOR__

```
#define __TIM4_UPD_OVF_VECTOR__ 23
```

irq23 - TIM4 Update/Overflow interrupt (shared with __TIM6_UPD_OVF_VECTOR__)

Definition at line 306 of file STM8AF_STM8S.h.

5.1.2.22 __TIM5_UPD_OVF_VECTOR__

```
#define __TIM5_UPD_OVF_VECTOR__ 13
```

irq13 - TIM5 Update/overflow interrupt (shared with __TIM2_UPD_OVF_VECTOR__)

Definition at line 266 of file STM8AF_STM8S.h.

5.1.2.23 __TLI_VECTOR__

```
#define __TLI_VECTOR__ 0
```

irq0 - External Top Level interrupt (TLI) for pin PD7

Definition at line 242 of file STM8AF_STM8S.h.

5.1.2.24 __UART1_RXF_VECTOR__

```
#define __UART1_RXF_VECTOR__ 18
```

irq18 - USART/UART1 receive (RX full) interrupt

Definition at line 283 of file STM8AF_STM8S.h.

5.1.2.25 __UART1_TXE_VECTOR__

```
#define __UART1_TXE_VECTOR__ 17
```

irq17 - USART/UART1 send (TX empty) interrupt

Definition at line 280 of file STM8AF_STM8S.h.

5.1.2.26 __UART2_RXF_VECTOR__

```
#define __UART2_RXF_VECTOR__ 21
```

irq21 - UART2 receive (RX full) interrupt (shared with __UART3_RXF_VECTOR__ and __UART4_RXF_VECTO↩R__)

Definition at line 294 of file STM8AF_STM8S.h.

5.1.2.27 __UART2_TXE_VECTOR__

```
#define __UART2_TXE_VECTOR__ 20
```

irq20 - UART2 send (TX empty) interrupt (shared with __UART3_TXE_VECTOR__ and __UART4_TXE_VECT↩OR__)

Definition at line 287 of file STM8AF_STM8S.h.

5.1.2.28 _ADC1

```
#define _ADC1 _SFR(ADC1_t, ADC1_AddressBase)
```

ADC1 struct/bit access.

Definition at line 4794 of file STM8AF_STM8S.h.

5.1.2.29 _ADC1_ADON

```
#define _ADC1_ADON ((uint8_t) (0x01 << 0))
```

ADC1 Conversion on/off [0] (in _ADC1_CR1)

Definition at line 4859 of file STM8AF_STM8S.h.

5.1.2.30 _ADC1_ALIGN

```
#define _ADC1_ALIGN ((uint8_t) (0x01 << 3))
```

ADC1 Data alignment [0] (in _ADC1_CR2)

Definition at line 4872 of file STM8AF_STM8S.h.

5.1.2.31 _ADC1_AWCRH

```
#define _ADC1_AWCRH _SFR(uint8_t, ADC1_AddressBase+0x2E)
```

ADC1 watchdog control register.

Definition at line 4830 of file STM8AF_STM8S.h.

5.1.2.32 _ADC1_AWCRH_RESET_VALUE

```
#define _ADC1_AWCRH_RESET_VALUE ((uint8_t) 0x00)
```

ADC1 watchdog control register reset value.

Definition at line 4844 of file STM8AF_STM8S.h.

5.1.2.33 _ADC1_AWCRL

```
#define _ADC1_AWCRL _SFR(uint8_t, ADC1_AddressBase+0x2F)
```

ADC1 watchdog control register.

Definition at line 4831 of file STM8AF_STM8S.h.

5.1.2.34 _ADC1_AWCRL_RESET_VALUE

```
#define _ADC1_AWCRL_RESET_VALUE ((uint8_t) 0x00)
```

ADC1 watchdog control register reset value.

Definition at line 4845 of file STM8AF_STM8S.h.

5.1.2.35 _ADC1_AWD

```
#define _ADC1_AWD ((uint8_t) (0x01 << 6))
```

ADC1 Analog Watchdog flag [0] (in _ADC1_CSR)

Definition at line 4855 of file STM8AF_STM8S.h.

5.1.2.36 _ADC1_AWDIE

```
#define _ADC1_AWDIE ((uint8_t) (0x01 << 4))
```

ADC1 Analog watchdog interrupt enable [0] (in _ADC1_CSR)

Definition at line 4853 of file STM8AF_STM8S.h.

5.1.2.37 _ADC1_AWSRH

```
#define _ADC1_AWSRH _SFR(uint8_t, ADC1_AddressBase+0x2C)
```

ADC1 watchdog status register.

Definition at line 4828 of file STM8AF_STM8S.h.

5.1.2.38 _ADC1_AWSRL

```
#define _ADC1_AWSRL _SFR(uint8_t, ADC1_AddressBase+0x2D)
```

ADC1 watchdog status register.

Definition at line 4829 of file STM8AF_STM8S.h.

5.1.2.39 _ADC1_CH

```
#define _ADC1_CH ((uint8_t) (0x0F << 0))
```

ADC1 Channel selection bits [3:0] (in _ADC1_CSR)

Definition at line 4848 of file STM8AF_STM8S.h.

5.1.2.40 _ADC1_CH0

```
#define _ADC1_CH0 ((uint8_t) (0x01 << 0))
```

ADC1 Channel selection bits [0] (in _ADC1_CSR)

Definition at line 4849 of file STM8AF_STM8S.h.

5.1.2.41 _ADC1_CH1

```
#define _ADC1_CH1 ((uint8_t) (0x01 << 1))
```

ADC1 Channel selection bits [1] (in _ADC1_CSR)

Definition at line 4850 of file STM8AF_STM8S.h.

5.1.2.42 _ADC1_CH2

```
#define _ADC1_CH2 ((uint8_t) (0x01 << 2))
```

ADC1 Channel selection bits [2] (in _ADC1_CSR)

Definition at line 4851 of file STM8AF_STM8S.h.

5.1.2.43 _ADC1_CH3

```
#define _ADC1_CH3 ((uint8_t) (0x01 << 3))
```

ADC1 Channel selection bits [3] (in _ADC1_CSR)

Definition at line 4852 of file STM8AF_STM8S.h.

5.1.2.44 _ADC1_CONT

```
#define _ADC1_CONT ((uint8_t) (0x01 << 1))
```

ADC1 Continuous conversion [0] (in _ADC1_CR1)

Definition at line 4860 of file STM8AF_STM8S.h.

5.1.2.45 _ADC1_CR1

```
#define _ADC1_CR1 _SFR(uint8_t, ADC1_AddressBase+0x21)
```

ADC1 Configuration Register 1.

Definition at line 4817 of file STM8AF_STM8S.h.

5.1.2.46 _ADC1_CR1_RESET_VALUE

```
#define _ADC1_CR1_RESET_VALUE ((uint8_t) 0x00)
```

ADC1 Configuration Register 1 reset value.

Definition at line 4835 of file STM8AF_STM8S.h.

5.1.2.47 _ADC1_CR2

```
#define _ADC1_CR2 _SFR(uint8_t, ADC1_AddressBase+0x22)
```

ADC1 Configuration Register 2.

Definition at line 4818 of file STM8AF_STM8S.h.

5.1.2.48 _ADC1_CR2_RESET_VALUE

```
#define _ADC1_CR2_RESET_VALUE ((uint8_t) 0x00)
```

ADC1 Configuration Register 2 reset value.

Definition at line 4836 of file STM8AF_STM8S.h.

5.1.2.49 _ADC1_CR3

```
#define _ADC1_CR3 _SFR(uint8_t, ADC1_AddressBase+0x23)
```

ADC1 Configuration Register 3.

Definition at line 4819 of file STM8AF_STM8S.h.

5.1.2.50 _ADC1_CR3_RESET_VALUE

```
#define _ADC1_CR3_RESET_VALUE ((uint8_t) 0x00)
```

ADC1 Configuration Register 3 reset value.

Definition at line 4837 of file STM8AF_STM8S.h.

5.1.2.51 _ADC1_CSR

```
#define _ADC1_CSR _SFR(uint8_t, ADC1_AddressBase+0x20)
```

ADC1 control/status register.

Definition at line 4816 of file STM8AF_STM8S.h.

5.1.2.52 _ADC1_CSR_RESET_VALUE

```
#define _ADC1_CSR_RESET_VALUE ((uint8_t) 0x00)
```

ADC1 control/status register reset value.

Definition at line 4834 of file STM8AF_STM8S.h.

5.1.2.53 _ADC1_DB0RH

```
#define _ADC1_DB0RH _SFR(uint8_t, ADC1_AddressBase+0x00)
```

ADC1 10-bit Data Buffer Register 0.

Definition at line 4795 of file STM8AF_STM8S.h.

5.1.2.54 _ADC1_DB0RL

```
#define _ADC1_DB0RL _SFR(uint8_t, ADC1_AddressBase+0x01)
```

ADC1 10-bit Data Buffer Register 0.

Definition at line 4796 of file STM8AF_STM8S.h.

5.1.2.55 _ADC1_DB1RH

```
#define _ADC1_DB1RH _SFR(uint8_t, ADC1_AddressBase+0x02)
```

ADC1 10-bit Data Buffer Register 1.

Definition at line 4797 of file STM8AF_STM8S.h.

5.1.2.56 _ADC1_DB1RL

```
#define _ADC1_DB1RL _SFR(uint8_t, ADC1_AddressBase+0x03)
```

ADC1 10-bit Data Buffer Register 1.

Definition at line 4798 of file STM8AF_STM8S.h.

5.1.2.57 _ADC1_DB2RH

```
#define _ADC1_DB2RH _SFR(uint8_t, ADC1_AddressBase+0x04)
```

ADC1 10-bit Data Buffer Register 2.

Definition at line 4799 of file STM8AF_STM8S.h.

5.1.2.58 _ADC1_DB2RL

```
#define _ADC1_DB2RL _SFR(uint8_t, ADC1_AddressBase+0x05)
```

ADC1 10-bit Data Buffer Register 2.

Definition at line 4800 of file STM8AF_STM8S.h.

5.1.2.59 _ADC1_DB3RH

```
#define _ADC1_DB3RH _SFR(uint8_t, ADC1_AddressBase+0x06)
```

ADC1 10-bit Data Buffer Register 3.

Definition at line 4801 of file STM8AF_STM8S.h.

5.1.2.60 _ADC1_DB3RL

```
#define _ADC1_DB3RL _SFR(uint8_t, ADC1_AddressBase+0x07)
```

ADC1 10-bit Data Buffer Register 3.

Definition at line 4802 of file STM8AF_STM8S.h.

5.1.2.61 _ADC1_DB4RH

```
#define _ADC1_DB4RH _SFR(uint8_t, ADC1_AddressBase+0x08)
```

ADC1 10-bit Data Buffer Register 4.

Definition at line 4803 of file STM8AF_STM8S.h.

5.1.2.62 _ADC1_DB4RL

```
#define _ADC1_DB4RL _SFR(uint8_t, ADC1_AddressBase+0x09)
```

ADC1 10-bit Data Buffer Register 4.

Definition at line 4804 of file STM8AF_STM8S.h.

5.1.2.63 _ADC1_DB5RH

```
#define _ADC1_DB5RH _SFR(uint8_t, ADC1_AddressBase+0x0A)
```

ADC1 10-bit Data Buffer Register 5.

Definition at line 4805 of file STM8AF_STM8S.h.

5.1.2.64 _ADC1_DB5RL

```
#define _ADC1_DB5RL _SFR(uint8_t, ADC1_AddressBase+0x0B)
```

ADC1 10-bit Data Buffer Register 5.

Definition at line 4806 of file STM8AF_STM8S.h.

5.1.2.65 _ADC1_DB6RH

```
#define _ADC1_DB6RH _SFR(uint8_t, ADC1_AddressBase+0x0C)
```

ADC1 10-bit Data Buffer Register 6.

Definition at line 4807 of file STM8AF_STM8S.h.

5.1.2.66 _ADC1_DB6RL

```
#define _ADC1_DB6RL _SFR(uint8_t, ADC1_AddressBase+0x0D)
```

ADC1 10-bit Data Buffer Register 6.

Definition at line 4808 of file STM8AF_STM8S.h.

5.1.2.67 _ADC1_DB7RH

```
#define _ADC1_DB7RH _SFR(uint8_t, ADC1_AddressBase+0x0E)
```

ADC1 10-bit Data Buffer Register 7.

Definition at line 4809 of file STM8AF_STM8S.h.

5.1.2.68 _ADC1_DB7RL

```
#define _ADC1_DB7RL _SFR(uint8_t, ADC1_AddressBase+0x0F)
```

ADC1 10-bit Data Buffer Register 7.

Definition at line 4810 of file STM8AF_STM8S.h.

5.1.2.69 _ADC1_DB8RH

```
#define _ADC1_DB8RH _SFR(uint8_t, ADC1_AddressBase+0x10)
```

ADC1 10-bit Data Buffer Register 8.

Definition at line 4811 of file STM8AF_STM8S.h.

5.1.2.70 _ADC1_DB8RL

```
#define _ADC1_DB8RL _SFR(uint8_t, ADC1_AddressBase+0x11)
```

ADC1 10-bit Data Buffer Register 8.

Definition at line 4812 of file STM8AF_STM8S.h.

5.1.2.71 _ADC1_DB9RH

```
#define _ADC1_DB9RH _SFR(uint8_t, ADC1_AddressBase+0x12)
```

ADC1 10-bit Data Buffer Register 9.

Definition at line 4813 of file STM8AF_STM8S.h.

5.1.2.72 _ADC1_DB9RL

```
#define _ADC1_DB9RL _SFR(uint8_t, ADC1_AddressBase+0x13)
```

ADC1 10-bit Data Buffer Register 9.

Definition at line 4814 of file STM8AF_STM8S.h.

5.1.2.73 _ADC1_DBUF

```
#define _ADC1_DBUF ((uint8_t) (0x01 << 7))
```

ADC1 Data buffer enable [0] (in _ADC1_CR3)

Definition at line 4882 of file STM8AF_STM8S.h.

5.1.2.74 _ADC1_DRH

```
#define _ADC1_DRH _SFR(uint8_t, ADC1_AddressBase+0x24)
```

ADC1 (unbuffered) 10-bit measurement result.

Definition at line 4820 of file STM8AF_STM8S.h.

5.1.2.75 _ADC1_DRL

```
#define _ADC1_DRL _SFR(uint8_t, ADC1_AddressBase+0x25)
```

ADC1 (unbuffered) 10-bit measurement result.

Definition at line 4821 of file STM8AF_STM8S.h.

5.1.2.76 _ADC1_EOC

```
#define _ADC1_EOC ((uint8_t) (0x01 << 7))
```

ADC1 End of conversion [0] (in _ADC1_CSR)

Definition at line 4856 of file STM8AF_STM8S.h.

5.1.2.77 _ADC1_EOCIE

```
#define _ADC1_EOCIE ((uint8_t) (0x01 << 5))
```

ADC1 Interrupt enable for EOC [0] (in _ADC1_CSR)

Definition at line 4854 of file STM8AF_STM8S.h.

5.1.2.78 _ADC1_EXTSEL

```
#define _ADC1_EXTSEL ((uint8_t) (0x03 << 4)
```

ADC1 External event selection [1:0] (in _ADC1_CR2)

Definition at line 4873 of file STM8AF_STM8S.h.

5.1.2.79 _ADC1_EXTSEL0

```
#define _ADC1_EXTSEL0 ((uint8_t) (0x01 << 4)
```

ADC1 External event selection [0] (in _ADC1_CR2)

Definition at line 4874 of file STM8AF_STM8S.h.

5.1.2.80 _ADC1_EXTSEL1

```
#define _ADC1_EXTSEL1 ((uint8_t) (0x01 << 5)
```

ADC1 External event selection [1] (in _ADC1_CR2)

Definition at line 4875 of file STM8AF_STM8S.h.

5.1.2.81 _ADC1_EXTTRIG

```
#define _ADC1_EXTTRIG ((uint8_t) (0x01 << 6))
```

ADC1 External trigger enable [0] (in _ADC1_CR2)

Definition at line 4876 of file STM8AF_STM8S.h.

5.1.2.82 _ADC1_HTRH

```
#define _ADC1_HTRH _SFR(uint8_t, ADC1_AddressBase+0x28)
```

ADC1 watchdog high threshold register.

Definition at line 4824 of file STM8AF_STM8S.h.

5.1.2.83 _ADC1_HTRH_RESET_VALUE

```
#define _ADC1_HTRH_RESET_VALUE ((uint8_t) 0xFF)
```

ADC1 watchdog high threshold register reset value.

Definition at line 4840 of file STM8AF_STM8S.h.

5.1.2.84 _ADC1_HTRL

```
#define _ADC1_HTRL _SFR(uint8_t, ADC1_AddressBase+0x29)
```

ADC1 watchdog high threshold register.

Definition at line 4825 of file STM8AF_STM8S.h.

5.1.2.85 _ADC1_HTRL_RESET_VALUE

```
#define _ADC1_HTRL_RESET_VALUE ((uint8_t) 0x03)
```

ADC1 watchdog high threshold register reset value.

Definition at line 4841 of file STM8AF_STM8S.h.

5.1.2.86 _ADC1_LTRH

```
#define _ADC1_LTRH _SFR(uint8_t, ADC1_AddressBase+0x2A)
```

ADC1 watchdog low threshold register.

Definition at line 4826 of file STM8AF_STM8S.h.

5.1.2.87 _ADC1_LTRH_RESET_VALUE

```
#define _ADC1_LTRH_RESET_VALUE ((uint8_t) 0x00)
```

ADC1 watchdog low threshold register reset value.

Definition at line 4842 of file STM8AF_STM8S.h.

5.1.2.88 _ADC1_LTRL

```
#define _ADC1_LTRL _SFR(uint8_t, ADC1_AddressBase+0x2B)
```

ADC1 watchdog low threshold register.

Definition at line 4827 of file STM8AF_STM8S.h.

5.1.2.89 _ADC1_LTRL_RESET_VALUE

```
#define _ADC1_LTRL_RESET_VALUE ((uint8_t) 0x00)
```

ADC1 watchdog low threshold register reset value.

Definition at line 4843 of file STM8AF_STM8S.h.

5.1.2.90 _ADC1_OVR

```
#define _ADC1_OVR ((uint8_t) (0x01 << 6))
```

ADC1 Overrun flag [0] (in _ADC1_CR3)

Definition at line 4881 of file STM8AF_STM8S.h.

5.1.2.91 _ADC1_SCAN

```
#define _ADC1_SCAN ((uint8_t) (0x01 << 1))
```

ADC1 Scan mode enable [0] (in _ADC1_CR2)

Definition at line 4870 of file STM8AF_STM8S.h.

5.1.2.92 _ADC1_SPSEL

```
#define _ADC1_SPSEL ((uint8_t) (0x07 << 4)
```

ADC1 clock prescaler selection [2:0] (in _ADC1_CR1)

Definition at line 4862 of file STM8AF_STM8S.h.

5.1.2.93 _ADC1_SPSEL0

```
#define _ADC1_SPSEL0 ((uint8_t) (0x01 << 4)
```

ADC1 clock prescaler selection [0] (in _ADC1_CR1)

Definition at line 4863 of file STM8AF_STM8S.h.

5.1.2.94 _ADC1_SPSEL1

```
#define _ADC1_SPSEL1 ((uint8_t) (0x01 << 5)
```

ADC1 clock prescaler selection [1] (in _ADC1_CR1)

Definition at line 4864 of file STM8AF_STM8S.h.

5.1.2.95 _ADC1_SPSEL2

```
#define _ADC1_SPSEL2 ((uint8_t) (0x01 << 6)
```

ADC1 clock prescaler selection [2] (in _ADC1_CR1)

Definition at line 4865 of file STM8AF_STM8S.h.

5.1.2.96 _ADC1_TDRH

```
#define _ADC1_TDRH _SFR(uint8_t, ADC1_AddressBase+0x26)
```

ADC1 Schmitt trigger disable register.

Definition at line 4822 of file STM8AF_STM8S.h.

5.1.2.97 _ADC1_TDRH_RESET_VALUE

```
#define _ADC1_TDRH_RESET_VALUE ((uint8_t) 0x00)
```

ADC1 Schmitt trigger disable register reset value.

Definition at line 4838 of file STM8AF_STM8S.h.

5.1.2.98 _ADC1_TDRL

```
#define _ADC1_TDRL _SFR(uint8_t, ADC1_AddressBase+0x27)
```

ADC1 Schmitt trigger disable register.

Definition at line 4823 of file STM8AF_STM8S.h.

5.1.2.99 _ADC1_TDRL_RESET_VALUE

```
#define _ADC1_TDRL_RESET_VALUE ((uint8_t) 0x00)
```

ADC1 Schmitt trigger disable register reset value.

Definition at line 4839 of file STM8AF_STM8S.h.

5.1.2.100 _ADC2

```
#define _ADC2 _SFR(ADC2_t, ADC2_AddressBase)
```

ADC2 struct/bit access.

Definition at line 4957 of file STM8AF_STM8S.h.

5.1.2.101 _ADC2_ADON

```
#define _ADC2_ADON ((uint8_t) (0x01 << 0))
```

ADC2 Conversion on/off [0] (in _ADC2_CR1)

Definition at line 4986 of file STM8AF_STM8S.h.

5.1.2.102 _ADC2_ALIGN

```
#define _ADC2_ALIGN ((uint8_t) (0x01 << 3))
```

ADC2 Data alignment [0] (in _ADC2_CR2)

Definition at line 4997 of file STM8AF_STM8S.h.

5.1.2.103 _ADC2_CH

```
#define _ADC2_CH ((uint8_t) (0x0F << 0))
```

ADC2 Channel selection bits [3:0] (in _ADC2_CSR)

Definition at line 4975 of file STM8AF_STM8S.h.

5.1.2.104 _ADC2_CH0

```
#define _ADC2_CH0 ((uint8_t) (0x01 << 0))
```

ADC2 Channel selection bits [0] (in _ADC2_CSR)

Definition at line 4976 of file STM8AF_STM8S.h.

5.1.2.105 _ADC2_CH1

```
#define _ADC2_CH1 ((uint8_t) (0x01 << 1))
```

ADC2 Channel selection bits [1] (in _ADC2_CSR)

Definition at line 4977 of file STM8AF_STM8S.h.

5.1.2.106 _ADC2_CH2

```
#define _ADC2_CH2 ((uint8_t) (0x01 << 2))
```

ADC2 Channel selection bits [2] (in _ADC2_CSR)

Definition at line 4978 of file STM8AF_STM8S.h.

5.1.2.107 _ADC2_CH3

```
#define _ADC2_CH3 ((uint8_t) (0x01 << 3))
```

ADC2 Channel selection bits [3] (in _ADC2_CSR)

Definition at line 4979 of file STM8AF_STM8S.h.

5.1.2.108 _ADC2_CONT

```
#define _ADC2_CONT ((uint8_t) (0x01 << 1))
```

ADC2 Continuous conversion [0] (in _ADC2_CR1)

Definition at line 4987 of file STM8AF_STM8S.h.

5.1.2.109 _ADC2_CR1

```
#define _ADC2_CR1 _SFR(uint8_t, ADC2_AddressBase+0x01)
```

ADC2 Configuration Register 1.

Definition at line 4959 of file STM8AF_STM8S.h.

5.1.2.110 _ADC2_CR1_RESET_VALUE

```
#define _ADC2_CR1_RESET_VALUE ((uint8_t) 0x00)
```

ADC2 Configuration Register 1 reset value.

Definition at line 4969 of file STM8AF_STM8S.h.

5.1.2.111 _ADC2_CR2

```
#define _ADC2_CR2 _SFR(uint8_t, ADC2_AddressBase+0x02)
```

ADC2 Configuration Register 2.

Definition at line 4960 of file STM8AF_STM8S.h.

5.1.2.112 _ADC2_CR2_RESET_VALUE

```
#define _ADC2_CR2_RESET_VALUE ((uint8_t) 0x00)
```

ADC2 Configuration Register 2 reset value.

Definition at line 4970 of file STM8AF_STM8S.h.

5.1.2.113 _ADC2_CSR

```
#define _ADC2_CSR _SFR(uint8_t, ADC2_AddressBase+0x00)
```

ADC2 control/status register.

Definition at line 4958 of file STM8AF_STM8S.h.

5.1.2.114 _ADC2_CSR_RESET_VALUE

```
#define _ADC2_CSR_RESET_VALUE ((uint8_t) 0x00)
```

ADC2 control/status register reset value.

Definition at line 4968 of file STM8AF_STM8S.h.

5.1.2.115 _ADC2_DRH

```
#define _ADC2_DRH _SFR(uint8_t, ADC2_AddressBase+0x04)
```

ADC2 (unbuffered) 10-bit measurement result.

Definition at line 4962 of file STM8AF_STM8S.h.

5.1.2.116 _ADC2_DRL

```
#define _ADC2_DRL _SFR(uint8_t, ADC2_AddressBase+0x05)
```

ADC2 (unbuffered) 10-bit measurement result.

Definition at line 4963 of file STM8AF_STM8S.h.

5.1.2.117 _ADC2_EOC

```
#define _ADC2_EOC ((uint8_t) (0x01 << 7))
```

ADC2 End of conversion [0] (in _ADC2_CSR)

Definition at line 4983 of file STM8AF_STM8S.h.

5.1.2.118 _ADC2_EOCIE

```
#define _ADC2_EOCIE ((uint8_t) (0x01 << 5))
```

ADC2 Interrupt enable for EOC [0] (in _ADC2_CSR)

Definition at line 4981 of file STM8AF_STM8S.h.

5.1.2.119 _ADC2_EXTSEL

```
#define _ADC2_EXTSEL ((uint8_t) (0x03 << 4)
```

ADC2 External event selection [1:0] (in _ADC2_CR2)

Definition at line 4998 of file STM8AF_STM8S.h.

5.1.2.120 _ADC2_EXTSEL0

```
#define _ADC2_EXTSEL0 ((uint8_t) (0x01 << 4)
```

ADC2 External event selection [0] (in _ADC2_CR2)

Definition at line 4999 of file STM8AF_STM8S.h.

5.1.2.121 _ADC2_EXTSEL1

```
#define _ADC2_EXTSEL1 ((uint8_t) (0x01 << 5)
```

ADC2 External event selection [1] (in _ADC2_CR2)

Definition at line 5000 of file STM8AF_STM8S.h.

5.1.2.122 _ADC2_EXTTRIG

```
#define _ADC2_EXTTRIG ((uint8_t) (0x01 << 6))
```

ADC2 External trigger enable [0] (in _ADC2_CR2)

Definition at line 5001 of file STM8AF_STM8S.h.

5.1.2.123 _ADC2_SPSEL

```
#define _ADC2_SPSEL ((uint8_t) (0x07 << 4)
```

ADC2 clock prescaler selection [2:0] (in _ADC2_CR1)

Definition at line 4989 of file STM8AF_STM8S.h.

5.1.2.124 _ADC2_SPSEL0

```
#define _ADC2_SPSEL0 ((uint8_t) (0x01 << 4)
```

ADC2 clock prescaler selection [0] (in _ADC2_CR1)

Definition at line 4990 of file STM8AF_STM8S.h.

5.1.2.125 _ADC2_SPSEL1

```
#define _ADC2_SPSEL1 ((uint8_t) (0x01 << 5)
```

ADC2 clock prescaler selection [1] (in _ADC2_CR1)

Definition at line 4991 of file STM8AF_STM8S.h.

5.1.2.126 _ADC2_SPSEL2

```
#define _ADC2_SPSEL2 ((uint8_t) (0x01 << 6)
```

ADC2 clock prescaler selection [2] (in _ADC2_CR1)

Definition at line 4992 of file STM8AF_STM8S.h.

5.1.2.127 _ADC2_TDRH

```
#define _ADC2_TDRH _SFR(uint8_t, ADC2_AddressBase+0x06)
```

ADC2 Schmitt trigger disable register.

Definition at line 4964 of file STM8AF_STM8S.h.

5.1.2.128 _ADC2_TDRH_RESET_VALUE

```
#define _ADC2_TDRH_RESET_VALUE ((uint8_t) 0x00)
```

ADC2 Schmitt trigger disable register reset value.

Definition at line 4972 of file STM8AF_STM8S.h.

5.1.2.129 _ADC2_TDRL

```
#define _ADC2_TDRL _SFR(uint8_t, ADC2_AddressBase+0x07)
```

ADC2 Schmitt trigger disable register.

Definition at line 4965 of file STM8AF_STM8S.h.

5.1.2.130 _ADC2_TDRL_RESET_VALUE

```
#define _ADC2_TDRL_RESET_VALUE ((uint8_t) 0x00)
```

ADC2 Schmitt trigger disable register reset value.

Definition at line 4971 of file STM8AF_STM8S.h.

5.1.2.131 _AWU

```
#define _AWU _SFR(AWU_t, AWU_AddressBase)
```

Auto Wake-Up struct/bit access.

Definition at line 1128 of file STM8AF_STM8S.h.

5.1.2.132 _AWU_APR

```
#define _AWU_APR _SFR(uint8_t, AWU_AddressBase+0x01)
```

Auto Wake-Up Asynchronous prescaler register (AWU_APR)

Definition at line 1130 of file STM8AF_STM8S.h.

5.1.2.133 _AWU_APR_RESET_VALUE

```
#define _AWU_APR_RESET_VALUE ((uint8_t) 0x3F)
```

Auto Wake-Up Asynchronous prescaler register reset value.

Definition at line 1135 of file STM8AF_STM8S.h.

5.1.2.134 _AWU_APRE

```
#define _AWU_APRE ((uint8_t) (0x3F << 0))
```

Auto-wakeup asynchronous prescaler divider [5:0] (in _AWU_APR)

Definition at line 1146 of file STM8AF_STM8S.h.

5.1.2.135 _AWU_APRE0

```
#define _AWU_APRE0 ((uint8_t) (0x01 << 0))
```

Auto-wakeup asynchronous prescaler divider [0] (in _AWU_APR)

Definition at line 1147 of file STM8AF_STM8S.h.

5.1.2.136 _AWU_APRE1

```
#define _AWU_APRE1 ((uint8_t) (0x01 << 1))
```

Auto-wakeup asynchronous prescaler divider [1] (in _AWU_APR)

Definition at line 1148 of file STM8AF_STM8S.h.

5.1.2.137 _AWU_APRE2

```
#define _AWU_APRE2 ((uint8_t) (0x01 << 2))
```

Auto-wakeup asynchronous prescaler divider [2] (in _AWU_APR)

Definition at line 1149 of file STM8AF_STM8S.h.

5.1.2.138 _AWU_APRE3

```
#define _AWU_APRE3 ((uint8_t) (0x01 << 3))
```

Auto-wakeup asynchronous prescaler divider [3] (in _AWU_APR)

Definition at line 1150 of file STM8AF_STM8S.h.

5.1.2.139 _AWU_APRE4

```
#define _AWU_APRE4 ((uint8_t) (0x01 << 4))
```

Auto-wakeup asynchronous prescaler divider [4] (in _AWU_APR)

Definition at line 1151 of file STM8AF_STM8S.h.

5.1.2.140 _AWU_APRE5

```
#define _AWU_APRE5 ((uint8_t) (0x01 << 5))
```

Auto-wakeup asynchronous prescaler divider [5] (in _AWU_APR)

Definition at line 1152 of file STM8AF_STM8S.h.

5.1.2.141 _AWU_AWUEN

```
#define _AWU_AWUEN ((uint8_t) (0x01 << 4))
```

Auto-wakeup enable [0] (in _AWU_CSR)

Definition at line 1141 of file STM8AF_STM8S.h.

5.1.2.142 _AWU_AWUF

```
#define _AWU_AWUF ((uint8_t) (0x01 << 5))
```

Auto-wakeup status flag [0] (in _AWU_CSR)

Definition at line 1142 of file STM8AF_STM8S.h.

5.1.2.143 _AWU_AWUTB

```
#define _AWU_AWUTB ((uint8_t) (0x0F << 0))
```

Auto-wakeup timebase selection [3:0] (in _AWU_APR)

Definition at line 1156 of file STM8AF_STM8S.h.

5.1.2.144 _AWU_AWUTB0

```
#define _AWU_AWUTB0 ((uint8_t) (0x01 << 0))
```

Auto-wakeup timebase selection [0] (in _AWU_APR)

Definition at line 1157 of file STM8AF_STM8S.h.

5.1.2.145 _AWU_AWUTB1

```
#define _AWU_AWUTB1 ((uint8_t) (0x01 << 1))
```

Auto-wakeup timebase selection [1] (in _AWU_APR)

Definition at line 1158 of file STM8AF_STM8S.h.

5.1.2.146 _AWU_AWUTB2

```
#define _AWU_AWUTB2 ((uint8_t) (0x01 << 2))
```

Auto-wakeup timebase selection [2] (in _AWU_APR)

Definition at line 1159 of file STM8AF_STM8S.h.

5.1.2.147 _AWU_AWUTB3

```
#define _AWU_AWUTB3 ((uint8_t) (0x01 << 3))
```

Auto-wakeup timebase selection [3] (in _AWU_APR)

Definition at line 1160 of file STM8AF_STM8S.h.

5.1.2.148 _AWU_CSR

```
#define _AWU_CSR _SFR(uint8_t, AWU_AddressBase+0x00)
```

Auto Wake-Up Control/status register (AWU_CSR)

Definition at line 1129 of file STM8AF_STM8S.h.

5.1.2.149 _AWU_CSR_RESET_VALUE

```
#define _AWU_CSR_RESET_VALUE ((uint8_t) 0x00)
```

Auto Wake-Up Control/status register reset value.

Definition at line 1134 of file STM8AF_STM8S.h.

5.1.2.150 _AWU_MSR

```
#define _AWU_MSR ((uint8_t) (0x01 << 0))
```

Auto Wake-Up LSI measurement enable [0] (in _AWU_CSR)

Definition at line 1139 of file STM8AF_STM8S.h.

5.1.2.151 _AWU_TBR

```
#define _AWU_TBR _SFR(uint8_t, AWU_AddressBase+0x02)
```

Auto Wake-Up Timebase selection register (AWU_TBR)

Definition at line 1131 of file STM8AF_STM8S.h.

5.1.2.152 _AWU_TBR_RESET_VALUE

```
#define _AWU_TBR_RESET_VALUE ((uint8_t) 0x00)
```

Auto Wake-Up Timebase selection register reset value.

Definition at line 1136 of file STM8AF_STM8S.h.

5.1.2.153 _BEEP

```
#define _BEEP _SFR(BEEP_t, BEEP_AddressBase)
```

Beeper struct/bit access.

Definition at line 1185 of file STM8AF_STM8S.h.

5.1.2.154 _BEEP_BEEPDIV

```
#define _BEEP_BEEPDIV ((uint8_t) (0x1F << 0))
```

Beeper clock prescaler divider [4:0] (in _BEEP_CSR)

Definition at line 1192 of file STM8AF_STM8S.h.

5.1.2.155 _BEEP_BEEPDIV0

```
#define _BEEP_BEEPDIV0 ((uint8_t) (0x01 << 0))
```

Beeper clock prescaler divider [0] (in _BEEP_CSR)

Definition at line 1193 of file STM8AF_STM8S.h.

5.1.2.156 _BEEP_BEEPDIV1

```
#define _BEEP_BEEPDIV1 ((uint8_t) (0x01 << 1))
```

Beeper clock prescaler divider [1] (in _BEEP_CSR)

Definition at line 1194 of file STM8AF_STM8S.h.

5.1.2.157 _BEEP_BEEPDIV2

```
#define _BEEP_BEEPDIV2 ((uint8_t) (0x01 << 2))
```

Beeper clock prescaler divider [2] (in _BEEP_CSR)

Definition at line 1195 of file STM8AF_STM8S.h.

5.1.2.158 _BEEP_BEEPDIV3

```
#define _BEEP_BEEPDIV3 ((uint8_t) (0x01 << 3))
```

Beeper clock prescaler divider [3] (in _BEEP_CSR)

Definition at line 1196 of file STM8AF_STM8S.h.

5.1.2.159 _BEEP_BEEPDIV4

```
#define _BEEP_BEEPDIV4 ((uint8_t) (0x01 << 4))
```

Beeper clock prescaler divider [4] (in _BEEP_CSR)

Definition at line 1197 of file STM8AF_STM8S.h.

5.1.2.160 _BEEP_BEEPEN

```
#define _BEEP_BEEPEN ((uint8_t) (0x01 << 5))
```

Beeper enable [0] (in _BEEP_CSR)

Definition at line 1198 of file STM8AF_STM8S.h.

5.1.2.161 _BEEP_BEEPSEL

```
#define _BEEP_BEEPSEL ((uint8_t) (0x03 << 6))
```

Beeper frequency selection [1:0] (in _BEEP_CSR)

Definition at line 1199 of file STM8AF_STM8S.h.

5.1.2.162 _BEEP_BEEPSEL0

```
#define _BEEP_BEEPSEL0 ((uint8_t) (0x01 << 6))
```

Beeper frequency selection [0] (in _BEEP_CSR)

Definition at line 1200 of file STM8AF_STM8S.h.

5.1.2.163 _BEEP_BEEPSEL1

```
#define _BEEP_BEEPSEL1 ((uint8_t) (0x01 << 7))
```

Beeper frequency selection [1] (in _BEEP_CSR)

Definition at line 1201 of file STM8AF_STM8S.h.

5.1.2.164 _BEEP_CSR

```
#define _BEEP_CSR _SFR(uint8_t, BEEP_AddressBase+0x00)
```

Beeper control/status register (BEEP_CSR)

Definition at line 1186 of file STM8AF_STM8S.h.

5.1.2.165 _BEEP_CSR_RESET_VALUE

```
#define _BEEP_CSR_RESET_VALUE ((uint8_t) 0x1F)
```

Beeper control/status register reset value.

Definition at line 1189 of file STM8AF_STM8S.h.

5.1.2.166 _BITS

```
#define _BITS unsigned int
```

data type in bit structs (follow C90 standard)

Definition at line 177 of file STM8AF_STM8S.h.

5.1.2.167 _CAN

```
#define _CAN _SFR(CAN_t, CAN_AddressBase)
```

CAN struct/bit access.

Definition at line 5946 of file STM8AF_STM8S.h.

5.1.2.168 _CAN_ABOM

```
#define _CAN_ABOM ((uint8_t) (0x01 << 6))
```

CAN Channel Automatic Bus-Off Management [0] (in _CAN_MCR)

Definition at line 6095 of file STM8AF_STM8S.h.

5.1.2.169 _CAN_ABRQ

```
#define _CAN_ABRQ ((uint8_t) (0x01 << 1))
```

CAN Abort request for mailbox [0] (in _CAN_MCSR, page 0,1,5)

Definition at line 6163 of file STM8AF_STM8S.h.

5.1.2.170 _CAN_ALST

```
#define _CAN_ALST ((uint8_t) (0x01 << 4))
```

CAN Arbitration lost [0] (in _CAN_MCSR, page 0,1,5)

Definition at line 6166 of file STM8AF_STM8S.h.

5.1.2.171 _CAN_AWUM

```
#define _CAN_AWUM ((uint8_t) (0x01 << 5))
```

CAN Channel Automatic Wakeup Mode [0] (in _CAN_MCR)

Definition at line 6094 of file STM8AF_STM8S.h.

5.1.2.172 _CAN_BOFF

```
#define _CAN_BOFF ((uint8_t) (0x01 << 2))
```

CAN Bus off flag [0] (in _CAN_ESR, page 6)

Definition at line 6188 of file STM8AF_STM8S.h.

5.1.2.173 _CAN_BOFIE

```
#define _CAN_BOFIE ((uint8_t) (0x01 << 2))
```

CAN Bus-Off interrupt enable [0] (in _CAN_EIER, page 6)

Definition at line 6199 of file STM8AF_STM8S.h.

5.1.2.174 _CAN_BRP

```
#define _CAN_BRP ((uint8_t) (0x3F << 0))
```

CAN Baud rate prescaler [5:0] (in _CAN_BTR1, page 6)

Definition at line 6207 of file STM8AF_STM8S.h.

5.1.2.175 _CAN_BRP0

```
#define _CAN_BRP0 ((uint8_t) (0x01 << 0))
```

CAN Baud rate prescaler [0] (in _CAN_BTR1, page 6)

Definition at line 6208 of file STM8AF_STM8S.h.

5.1.2.176 _CAN_BRP1

```
#define _CAN_BRP1 ((uint8_t) (0x01 << 1))
```

CAN Baud rate prescaler [1] (in _CAN_BTR1, page 6)

Definition at line 6209 of file STM8AF_STM8S.h.

5.1.2.177 _CAN_BRP2

```
#define _CAN_BRP2 ((uint8_t) (0x01 << 2))
```

CAN Baud rate prescaler [2] (in _CAN_BTR1, page 6)

Definition at line 6210 of file STM8AF_STM8S.h.

5.1.2.178 _CAN_BRP3

```
#define _CAN_BRP3 ((uint8_t) (0x01 << 3))
```

CAN Baud rate prescaler [3] (in _CAN_BTR1, page 6)

Definition at line 6211 of file STM8AF_STM8S.h.

5.1.2.179 _CAN_BRP4

```
#define _CAN_BRP4 ((uint8_t) (0x01 << 4))
```

CAN Baud rate prescaler [4] (in _CAN_BTR1, page 6)

Definition at line 6212 of file STM8AF_STM8S.h.

5.1.2.180 _CAN_BRP5

```
#define _CAN_BRP5 ((uint8_t) (0x01 << 5))
```

CAN Baud rate prescaler [5] (in _CAN_BTR1, page 6)

Definition at line 6213 of file STM8AF_STM8S.h.

5.1.2.181 _CAN_BS1

```
#define _CAN_BS1 ((uint8_t) (0x0F << 0))
```

CAN Bit segment 1 [3:0] (in _CAN_BTR2, page 6)

Definition at line 6219 of file STM8AF_STM8S.h.

5.1.2.182 _CAN_BS10

```
#define _CAN_BS10 ((uint8_t) (0x01 << 0))
```

CAN Bit segment 1 [0] (in _CAN_BTR2, page 6)

Definition at line 6220 of file STM8AF_STM8S.h.

5.1.2.183 _CAN_BS11

```
#define _CAN_BS11 ((uint8_t) (0x01 << 1))
```

CAN Bit segment 1 [1] (in _CAN_BTR2, page 6)

Definition at line 6221 of file STM8AF_STM8S.h.

5.1.2.184 _CAN_BS12

```
#define _CAN_BS12 ((uint8_t) (0x01 << 2))
```

CAN Bit segment 1 [2] (in _CAN_BTR2, page 6)

Definition at line 6222 of file STM8AF_STM8S.h.

5.1.2.185 _CAN_BS13

```
#define _CAN_BS13 ((uint8_t) (0x01 << 3))
```

CAN Bit segment 1 [3] (in _CAN_BTR2, page 6)

Definition at line 6223 of file STM8AF_STM8S.h.

5.1.2.186 _CAN_BS2

```
#define _CAN_BS2 ((uint8_t) (0x07 << 4))
```

CAN Bit segment 2 [2:0] (in _CAN_BTR2, page 6)

Definition at line 6224 of file STM8AF_STM8S.h.

5.1.2.187 _CAN_BS20

```
#define _CAN_BS20 ((uint8_t) (0x01 << 4))
```

CAN Bit segment 2 [0] (in _CAN_BTR2, page 6)

Definition at line 6225 of file STM8AF_STM8S.h.

5.1.2.188 _CAN_BS21

```
#define _CAN_BS21 ((uint8_t) (0x01 << 5))
```

CAN Bit segment 2 [1] (in _CAN_BTR2, page 6)

Definition at line 6226 of file STM8AF_STM8S.h.

5.1.2.189 _CAN_BS22

```
#define _CAN_BS22 ((uint8_t) (0x01 << 6))
```

CAN Bit segment 2 [2] (in _CAN_BTR2, page 6)

Definition at line 6227 of file STM8AF_STM8S.h.

5.1.2.190 _CAN_BTR1

```
#define _CAN_BTR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)
```

CAN bit timing register 1 (page 6)

Definition at line 6037 of file STM8AF_STM8S.h.

5.1.2.191 _CAN_BTR1_RESET_VALUE

```
#define _CAN_BTR1_RESET_VALUE ((uint8_t) 0x40)
```

CAN bit timing register 1 (page 6) reset value.

Definition at line 6081 of file STM8AF_STM8S.h.

5.1.2.192 _CAN_BTR2

```
#define _CAN_BTR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)
```

CAN bit timing register 2 (page 6)

Definition at line 6038 of file STM8AF_STM8S.h.

5.1.2.193 _CAN_BTR2_RESET_VALUE

```
#define _CAN_BTR2_RESET_VALUE ((uint8_t) 0x23)
```

CAN bit timing register 2 (page 6) reset value.

Definition at line 6082 of file STM8AF_STM8S.h.

5.1.2.194 _CAN_CODE

```
#define _CAN_CODE ((uint8_t) (0x03 << 0))
```

CAN Mailbox Code [1:0] (in _CAN_TPR)

Definition at line 6118 of file STM8AF_STM8S.h.

5.1.2.195 _CAN_CODE0

```
#define _CAN_CODE0 ((uint8_t) (0x01 << 0))
```

CAN Mailbox Code [0] (in _CAN_TPR)

Definition at line 6119 of file STM8AF_STM8S.h.

5.1.2.196 _CAN_CODE1

```
#define _CAN_CODE1 ((uint8_t) (0x01 << 1))
```

CAN Mailbox Code [1] (in _CAN_TPR)

Definition at line 6120 of file STM8AF_STM8S.h.

5.1.2.197 _CAN_DGR

```
#define _CAN_DGR _SFR(uint8_t, CAN_AddressBase+0x06)
```

CAN diagnosis register.

Definition at line 5953 of file STM8AF_STM8S.h.

5.1.2.198 _CAN_DGR_RESET_VALUE

```
#define _CAN_DGR_RESET_VALUE ((uint8_t) 0x0C)
```

CAN diagnosis register reset value.

Definition at line 6072 of file STM8AF_STM8S.h.

5.1.2.199 _CAN_DLC

```
#define _CAN_DLC ((uint8_t) (0x0F << 0))
```

CAN Data length code [3:0] (in _CAN_MDLCR, page 0,1,5,7)

Definition at line 6171 of file STM8AF_STM8S.h.

5.1.2.200 _CAN_DLC0

```
#define _CAN_DLC0 ((uint8_t) (0x01 << 0))
```

CAN Data length code [0] (in _CAN_MDLCR, page 0,1,5,7)

Definition at line 6172 of file STM8AF_STM8S.h.

5.1.2.201 _CAN_DLC1

```
#define _CAN_DLC1 ((uint8_t) (0x01 << 1))
```

CAN Data length code [1] (in _CAN_MDLCR, page 0,1,5,7)

Definition at line 6173 of file STM8AF_STM8S.h.

5.1.2.202 _CAN_DLC2

```
#define _CAN_DLC2 ((uint8_t) (0x01 << 2))
```

CAN Data length code [2] (in _CAN_MDLCR, page 0,1,5,7)

Definition at line 6174 of file STM8AF_STM8S.h.

5.1.2.203 _CAN_DLC3

```
#define _CAN_DLC3 ((uint8_t) (0x01 << 3))
```

CAN Data length code [3] (in _CAN_MDLCR, page 0,1,5,7)

Definition at line 6175 of file STM8AF_STM8S.h.

5.1.2.204 _CAN_EIER

```
#define _CAN_EIER _SFR(uint8_t, CAN_AddressBase+0x08+0x01)
```

CAN error interrupt enable register (page 6)

Definition at line 6034 of file STM8AF_STM8S.h.

5.1.2.205 _CAN_EIER_RESET_VALUE

```
#define _CAN_EIER_RESET_VALUE ((uint8_t) 0x00)
```

CAN error interrupt enable register (page 6) reset value.

Definition at line 6078 of file STM8AF_STM8S.h.

5.1.2.206 _CAN_EPVF

```
#define _CAN_EPVF ((uint8_t) (0x01 << 1))
```

CAN Error passive flag [0] (in _CAN_ESR, page 6)

Definition at line 6187 of file STM8AF_STM8S.h.

5.1.2.207 _CAN_EPVIE

```
#define _CAN_EPVIE ((uint8_t) (0x01 << 1))
```

CAN Error passive interrupt enable [0] (in _CAN_EIER, page 6)

Definition at line 6198 of file STM8AF_STM8S.h.

5.1.2.208 _CAN_ERRI

```
#define _CAN_ERRI ((uint8_t) (0x01 << 2))
```

CAN Error Interrupt [0] (in _CAN_MSR)

Definition at line 6101 of file STM8AF_STM8S.h.

5.1.2.209 _CAN_ERRIE

```
#define _CAN_ERRIE ((uint8_t) (0x01 << 6))
```

CAN Error interrupt enable [0] (in _CAN_EIER, page 6)

Definition at line 6203 of file STM8AF_STM8S.h.

5.1.2.210 _CAN_ESR

```
#define _CAN_ESR _SFR(uint8_t, CAN_AddressBase+0x08+0x00)
```

CAN error status register (page 6)

Definition at line 6033 of file STM8AF_STM8S.h.

5.1.2.211 _CAN_ESR_RESET_VALUE

```
#define _CAN_ESR_RESET_VALUE ((uint8_t) 0x00)
```

CAN error status register (page 6) reset value.

Definition at line 6077 of file STM8AF_STM8S.h.

5.1.2.212 _CAN_EWGF

```
#define _CAN_EWGF ((uint8_t) (0x01 << 0))
```

CAN Error warning flag [0] (in _CAN_ESR, page 6)

Definition at line 6186 of file STM8AF_STM8S.h.

5.1.2.213 _CAN_EWGIE

```
#define _CAN_EWGIE ((uint8_t) (0x01 << 0))
```

CAN Error warning interrupt enable [0] (in _CAN_EIER, page 6)

Definition at line 6197 of file STM8AF_STM8S.h.

5.1.2.214 _CAN_F0R1

```
#define _CAN_F0R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x00)
```

CAN acceptance filter 0/1 (page 2)

Definition at line 5977 of file STM8AF_STM8S.h.

5.1.2.215 _CAN_F0R2

```
#define _CAN_F0R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x01)
```

CAN acceptance filter 0/2 (page 2)

Definition at line 5978 of file STM8AF_STM8S.h.

5.1.2.216 _CAN_F0R3

```
#define _CAN_F0R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x02)
```

CAN acceptance filter 0/3 (page 2)

Definition at line 5979 of file STM8AF_STM8S.h.

5.1.2.217 _CAN_F0R4

```
#define _CAN_F0R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x03)
```

CAN acceptance filter 0/4 (page 2)

Definition at line 5980 of file STM8AF_STM8S.h.

5.1.2.218 _CAN_F0R5

```
#define _CAN_F0R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)
```

CAN acceptance filter 0/5 (page 2)

Definition at line 5981 of file STM8AF_STM8S.h.

5.1.2.219 _CAN_F0R6

```
#define _CAN_F0R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)
```

CAN acceptance filter 0/6 (page 2)

Definition at line 5982 of file STM8AF_STM8S.h.

5.1.2.220 _CAN_F0R7

```
#define _CAN_F0R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x06)
```

CAN acceptance filter 0/7 (page 2)

Definition at line 5983 of file STM8AF_STM8S.h.

5.1.2.221 _CAN_F0R8

```
#define _CAN_F0R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x07)
```

CAN acceptance filter 0/8 (page 2)

Definition at line 5984 of file STM8AF_STM8S.h.

5.1.2.222 _CAN_F1R1

```
#define _CAN_F1R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)
```

CAN acceptance filter 1/1 (page 2)

Definition at line 5985 of file STM8AF_STM8S.h.

5.1.2.223 _CAN_F1R2

```
#define _CAN_F1R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)
```

CAN acceptance filter 1/2 (page 2)

Definition at line 5986 of file STM8AF_STM8S.h.

5.1.2.224 _CAN_F1R3

```
#define _CAN_F1R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)
```

CAN acceptance filter 1/3 (page 2)

Definition at line 5987 of file STM8AF_STM8S.h.

5.1.2.225 _CAN_F1R4

```
#define _CAN_F1R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)
```

CAN acceptance filter 1/4 (page 2)

Definition at line 5988 of file STM8AF_STM8S.h.

5.1.2.226 _CAN_F1R5

```
#define _CAN_F1R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)
```

CAN acceptance filter 1/5 (page 2)

Definition at line 5989 of file STM8AF_STM8S.h.

5.1.2.227 _CAN_F1R6

```
#define _CAN_F1R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x0D)
```

CAN acceptance filter 1/6 (page 2)

Definition at line 5990 of file STM8AF_STM8S.h.

5.1.2.228 _CAN_F1R7

```
#define _CAN_F1R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x0E)
```

CAN acceptance filter 1/7 (page 2)

Definition at line 5991 of file STM8AF_STM8S.h.

5.1.2.229 _CAN_F1R8

```
#define _CAN_F1R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x0F)
```

CAN acceptance filter 1/8 (page 2)

Definition at line 5992 of file STM8AF_STM8S.h.

5.1.2.230 _CAN_F2R1

```
#define _CAN_F2R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x00)
```

CAN acceptance filter 2/1 (page 3)

Definition at line 5995 of file STM8AF_STM8S.h.

5.1.2.231 _CAN_F2R2

```
#define _CAN_F2R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x01)
```

CAN acceptance filter 2/2 (page 3)

Definition at line 5996 of file STM8AF_STM8S.h.

5.1.2.232 _CAN_F2R3

```
#define _CAN_F2R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x02)
```

CAN acceptance filter 2/3 (page 3)

Definition at line 5997 of file STM8AF_STM8S.h.

5.1.2.233 _CAN_F2R4

```
#define _CAN_F2R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x03)
```

CAN acceptance filter 2/4 (page 3)

Definition at line 5998 of file STM8AF_STM8S.h.

5.1.2.234 _CAN_F2R5

```
#define _CAN_F2R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)
```

CAN acceptance filter 2/5 (page 3)

Definition at line 5999 of file STM8AF_STM8S.h.

5.1.2.235 _CAN_F2R6

```
#define _CAN_F2R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)
```

CAN acceptance filter 2/6 (page 3)

Definition at line 6000 of file STM8AF_STM8S.h.

5.1.2.236 _CAN_F2R7

```
#define _CAN_F2R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x06)
```

CAN acceptance filter 2/7 (page 3)

Definition at line 6001 of file STM8AF_STM8S.h.

5.1.2.237 _CAN_F2R8

```
#define _CAN_F2R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x07)
```

CAN acceptance filter 2/8 (page 3)

Definition at line 6002 of file STM8AF_STM8S.h.

5.1.2.238 _CAN_F3R1

```
#define _CAN_F3R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)
```

CAN acceptance filter 3/1 (page 3)

Definition at line 6003 of file STM8AF_STM8S.h.

5.1.2.239 _CAN_F3R2

```
#define _CAN_F3R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)
```

CAN acceptance filter 3/2 (page 3)

Definition at line 6004 of file STM8AF_STM8S.h.

5.1.2.240 _CAN_F3R3

```
#define _CAN_F3R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)
```

CAN acceptance filter 3/3 (page 3)

Definition at line 6005 of file STM8AF_STM8S.h.

5.1.2.241 _CAN_F3R4

```
#define _CAN_F3R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)
```

CAN acceptance filter 3/4 (page 3)

Definition at line 6006 of file STM8AF_STM8S.h.

5.1.2.242 _CAN_F3R5

```
#define _CAN_F3R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)
```

CAN acceptance filter 3/5 (page 3)

Definition at line 6007 of file STM8AF_STM8S.h.

5.1.2.243 _CAN_F3R6

```
#define _CAN_F3R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x0D)
```

CAN acceptance filter 3/6 (page 3)

Definition at line 6008 of file STM8AF_STM8S.h.

5.1.2.244 _CAN_F3R7

```
#define _CAN_F3R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x0E)
```

CAN acceptance filter 3/7 (page 3)

Definition at line 6009 of file STM8AF_STM8S.h.

5.1.2.245 _CAN_F3R8

```
#define _CAN_F3R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x0F)
```

CAN acceptance filter 3/8 (page 3)

Definition at line 6010 of file STM8AF_STM8S.h.

5.1.2.246 _CAN_F4R1

```
#define _CAN_F4R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x00)
```

CAN acceptance filter 4/1 (page 4)

Definition at line 6013 of file STM8AF_STM8S.h.

5.1.2.247 _CAN_F4R2

```
#define _CAN_F4R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x01)
```

CAN acceptance filter 4/2 (page 4)

Definition at line 6014 of file STM8AF_STM8S.h.

5.1.2.248 _CAN_F4R3

```
#define _CAN_F4R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x02)
```

CAN acceptance filter 4/3 (page 4)

Definition at line 6015 of file STM8AF_STM8S.h.

5.1.2.249 _CAN_F4R4

```
#define _CAN_F4R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x03)
```

CAN acceptance filter 4/4 (page 4)

Definition at line 6016 of file STM8AF_STM8S.h.

5.1.2.250 _CAN_F4R5

```
#define _CAN_F4R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)
```

CAN acceptance filter 4/5 (page 4)

Definition at line 6017 of file STM8AF_STM8S.h.

5.1.2.251 _CAN_F4R6

```
#define _CAN_F4R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)
```

CAN acceptance filter 4/6 (page 4)

Definition at line 6018 of file STM8AF_STM8S.h.

5.1.2.252 _CAN_F4R7

```
#define _CAN_F4R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x06)
```

CAN acceptance filter 4/7 (page 4)

Definition at line 6019 of file STM8AF_STM8S.h.

5.1.2.253 _CAN_F4R8

```
#define _CAN_F4R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x07)
```

CAN acceptance filter 4/8 (page 4)

Definition at line 6020 of file STM8AF_STM8S.h.

5.1.2.254 _CAN_F5R1

```
#define _CAN_F5R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)
```

CAN acceptance filter 5/1 (page 4)

Definition at line 6021 of file STM8AF_STM8S.h.

5.1.2.255 _CAN_F5R2

```
#define _CAN_F5R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)
```

CAN acceptance filter 5/2 (page 4)

Definition at line 6022 of file STM8AF_STM8S.h.

5.1.2.256 _CAN_F5R3

```
#define _CAN_F5R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)
```

CAN acceptance filter 5/3 (page 4)

Definition at line 6023 of file STM8AF_STM8S.h.

5.1.2.257 _CAN_F5R4

```
#define _CAN_F5R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)
```

CAN acceptance filter 5/4 (page 4)

Definition at line 6024 of file STM8AF_STM8S.h.

5.1.2.258 _CAN_F5R5

```
#define _CAN_F5R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)
```

CAN acceptance filter 5/5 (page 4)

Definition at line 6025 of file STM8AF_STM8S.h.

5.1.2.259 _CAN_F5R6

```
#define _CAN_F5R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x0D)
```

CAN acceptance filter 5/6 (page 4)

Definition at line 6026 of file STM8AF_STM8S.h.

5.1.2.260 _CAN_F5R7

```
#define _CAN_F5R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x0E)
```

CAN acceptance filter 5/7 (page 4)

Definition at line 6027 of file STM8AF_STM8S.h.

5.1.2.261 _CAN_F5R8

```
#define _CAN_F5R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x0F)
```

CAN acceptance filter 5/8 (page 4)

Definition at line 6028 of file STM8AF_STM8S.h.

5.1.2.262 _CAN_FACT0

```
#define _CAN_FACT0 ((uint8_t) (0x01 << 0))
```

CAN Filter 0 active [0] (in _CAN_FCR1, page 6)

Definition at line 6248 of file STM8AF_STM8S.h.

5.1.2.263 _CAN_FACT1

```
#define _CAN_FACT1 ((uint8_t) (0x01 << 4))
```

CAN Filter 1 active [0] (in _CAN_FCR1, page 6)

Definition at line 6253 of file STM8AF_STM8S.h.

5.1.2.264 _CAN_FACT2

```
#define _CAN_FACT2 ((uint8_t) (0x01 << 0))
```

CAN Filter 2 active [0] (in _CAN_FCR2, page 6)

Definition at line 6260 of file STM8AF_STM8S.h.

5.1.2.265 _CAN_FACT3

```
#define _CAN_FACT3 ((uint8_t) (0x01 << 4))
```

CAN Filter 3 active [0] (in _CAN_FCR2, page 6)

Definition at line 6265 of file STM8AF_STM8S.h.

5.1.2.266 _CAN_FACT4

```
#define _CAN_FACT4 ((uint8_t) (0x01 << 0))
```

CAN Filter 4 active [0] (in _CAN_FCR3, page 6)

Definition at line 6272 of file STM8AF_STM8S.h.

5.1.2.267 _CAN_FACT5

```
#define _CAN_FACT5 ((uint8_t) (0x01 << 4))
```

CAN Filter 5 active [0] (in _CAN_FCR2, page 6)

Definition at line 6277 of file STM8AF_STM8S.h.

5.1.2.268 _CAN_FCR1

```
#define _CAN_FCR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)
```

CAN filter configuration register 1 (page 6)

Definition at line 6042 of file STM8AF_STM8S.h.

5.1.2.269 _CAN_FCR2

```
#define _CAN_FCR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)
```

CAN filter configuration register 2 (page 6)

Definition at line 6043 of file STM8AF_STM8S.h.

5.1.2.270 _CAN_FCR3

```
#define _CAN_FCR3 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)
```

CAN filter configuration register 3 (page 6)

Definition at line 6044 of file STM8AF_STM8S.h.

5.1.2.271 _CAN_FCR_RESET_VALUE

```
#define _CAN_FCR_RESET_VALUE ((uint8_t) 0x00)
```

CAN filter configuration register reset value.

Definition at line 6085 of file STM8AF_STM8S.h.

5.1.2.272 _CAN_FFIE

```
#define _CAN_FFIE ((uint8_t) (0x01 << 2))
```

CAN FIFO Full Interrupt Enable [0] (in _CAN_IER)

Definition at line 6141 of file STM8AF_STM8S.h.

5.1.2.273 _CAN_FMH0

```
#define _CAN_FMH0 ((uint8_t) (0x01 << 1))
```

CAN Filter 0 mode high [0] (in _CAN_FMR1, page 6)

Definition at line 6232 of file STM8AF_STM8S.h.

5.1.2.274 _CAN_FMH1

```
#define _CAN_FMH1 ((uint8_t) (0x01 << 3))
```

CAN Filter 1 mode high [0] (in _CAN_FMR1, page 6)

Definition at line 6234 of file STM8AF_STM8S.h.

5.1.2.275 _CAN_FMH2

```
#define _CAN_FMH2 ((uint8_t) (0x01 << 5))
```

CAN Filter 2 mode high [0] (in _CAN_FMR1, page 6)

Definition at line 6236 of file STM8AF_STM8S.h.

5.1.2.276 _CAN_FMH3

```
#define _CAN_FMH3 ((uint8_t) (0x01 << 7))
```

CAN Filter 3 mode high [0] (in _CAN_FMR1, page 6)

Definition at line 6238 of file STM8AF_STM8S.h.

5.1.2.277 _CAN_FMH4

```
#define _CAN_FMH4 ((uint8_t) (0x01 << 1))
```

CAN Filter 4 mode high [0] (in _CAN_FMR2, page 6)

Definition at line 6242 of file STM8AF_STM8S.h.

5.1.2.278 _CAN_FMH5

```
#define _CAN_FMH5 ((uint8_t) (0x01 << 3))
```

CAN Filter 5 mode high [0] (in _CAN_FMR2, page 6)

Definition at line 6244 of file STM8AF_STM8S.h.

5.1.2.279 _CAN_FML0

```
#define _CAN_FML0 ((uint8_t) (0x01 << 0))
```

CAN Filter 0 mode low [0] (in _CAN_FMR1, page 6)

Definition at line 6231 of file STM8AF_STM8S.h.

5.1.2.280 _CAN_FML1

```
#define _CAN_FML1 ((uint8_t) (0x01 << 2))
```

CAN Filter 1 mode low [0] (in _CAN_FMR1, page 6)

Definition at line 6233 of file STM8AF_STM8S.h.

5.1.2.281 _CAN_FML2

```
#define _CAN_FML2 ((uint8_t) (0x01 << 4))
```

CAN Filter 2 mode low [0] (in _CAN_FMR1, page 6)

Definition at line 6235 of file STM8AF_STM8S.h.

5.1.2.282 _CAN_FML3

```
#define _CAN_FML3 ((uint8_t) (0x01 << 6))
```

CAN Filter 3 mode low [0] (in _CAN_FMR1, page 6)

Definition at line 6237 of file STM8AF_STM8S.h.

5.1.2.283 _CAN_FML4

```
#define _CAN_FML4 ((uint8_t) (0x01 << 0))
```

CAN Filter 4 mode low [0] (in _CAN_FMR2, page 6)

Definition at line 6241 of file STM8AF_STM8S.h.

5.1.2.284 _CAN_FML5

```
#define _CAN_FML5 ((uint8_t) (0x01 << 2))
```

CAN Filter 5 mode low [0] (in _CAN_FMR2, page 6)

Definition at line 6243 of file STM8AF_STM8S.h.

5.1.2.285 _CAN_FMP

```
#define _CAN_FMP ((uint8_t) (0x03 << 0))
```

CAN FIFO Message Pending [1:0] (in _CAN_RFR)

Definition at line 6129 of file STM8AF_STM8S.h.

5.1.2.286 _CAN_FMP0

```
#define _CAN_FMP0 ((uint8_t) (0x01 << 0))
```

CAN FIFO Message Pending [0] (in _CAN_RFR)

Definition at line 6130 of file STM8AF_STM8S.h.

5.1.2.287 _CAN_FMP1

```
#define _CAN_FMP1 ((uint8_t) (0x01 << 1))
```

CAN FIFO Message Pending [1] (in _CAN_RFR)

Definition at line 6131 of file STM8AF_STM8S.h.

5.1.2.288 _CAN_FMPIE

```
#define _CAN_FMPIE ((uint8_t) (0x01 << 1))
```

CAN FIFO Message Pending Interrupt Enable [0] (in _CAN_IER)

Definition at line 6140 of file STM8AF_STM8S.h.

5.1.2.289 _CAN_FMR1

```
#define _CAN_FMR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)
```

CAN filter mode register 1 (page 6)

Definition at line 6040 of file STM8AF_STM8S.h.

5.1.2.290 _CAN_FMR1_RESET_VALUE

```
#define _CAN_FMR1_RESET_VALUE ((uint8_t) 0x00)
```

CAN filter mode register 1 (page 6) reset value.

Definition at line 6083 of file STM8AF_STM8S.h.

5.1.2.291 _CAN_FMR2

```
#define _CAN_FMR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)
```

CAN filter mode register 2 (page 6)

Definition at line 6041 of file STM8AF_STM8S.h.

5.1.2.292 _CAN_FMR2_RESET_VALUE

```
#define _CAN_FMR2_RESET_VALUE ((uint8_t) 0x00)
```

CAN filter mode register 2 (page 6) reset value.

Definition at line 6084 of file STM8AF_STM8S.h.

5.1.2.293 _CAN_FOVIE

```
#define _CAN_FOVIE ((uint8_t) (0x01 << 3))
```

CAN FIFO Overrun Interrupt Enable [0] (in _CAN_IER)

Definition at line 6142 of file STM8AF_STM8S.h.

5.1.2.294 _CAN_FOVR

```
#define _CAN_FOVR ((uint8_t) (0x01 << 4))
```

CAN FIFO Overrun [0] (in _CAN_RFR)

Definition at line 6134 of file STM8AF_STM8S.h.

5.1.2.295 _CAN_FSC0

```
#define _CAN_FSC0 ((uint8_t) (0x03 << 1))
```

CAN Filter 0 scale configuration [1:0] (in _CAN_FCR1, page 6)

Definition at line 6249 of file STM8AF_STM8S.h.

5.1.2.296 _CAN_FSC00

```
#define _CAN_FSC00 ((uint8_t) (0x01 << 1))
```

CAN Filter 0 scale configuration [0] (in _CAN_FCR1, page 6)

Definition at line 6250 of file STM8AF_STM8S.h.

5.1.2.297 _CAN_FSC01

```
#define _CAN_FSC01 ((uint8_t) (0x01 << 2))
```

CAN Filter 0 scale configuration [1] (in _CAN_FCR1, page 6)

Definition at line 6251 of file STM8AF_STM8S.h.

5.1.2.298 _CAN_FSC1

```
#define _CAN_FSC1 ((uint8_t) (0x03 << 5))
```

CAN Filter 1 scale configuration [1:0] (in _CAN_FCR1, page 6)

Definition at line 6254 of file STM8AF_STM8S.h.

5.1.2.299 _CAN_FSC10

```
#define _CAN_FSC10 ((uint8_t) (0x01 << 5))
```

CAN Filter 1 scale configuration [0] (in _CAN_FCR1, page 6)

Definition at line 6255 of file STM8AF_STM8S.h.

5.1.2.300 _CAN_FSC11

```
#define _CAN_FSC11 ((uint8_t) (0x01 << 6))
```

CAN Filter 1 scale configuration [1] (in _CAN_FCR1, page 6)

Definition at line 6256 of file STM8AF_STM8S.h.

5.1.2.301 _CAN_FSC2

```
#define _CAN_FSC2 ((uint8_t) (0x03 << 1))
```

CAN Filter 2 scale configuration [1:0] (in _CAN_FCR2, page 6)

Definition at line 6261 of file STM8AF_STM8S.h.

5.1.2.302 _CAN_FSC20

```
#define _CAN_FSC20 ((uint8_t) (0x01 << 1))
```

CAN Filter 2 scale configuration [0] (in _CAN_FCR2, page 6)

Definition at line 6262 of file STM8AF_STM8S.h.

5.1.2.303 _CAN_FSC21

```
#define _CAN_FSC21 ((uint8_t) (0x01 << 2))
```

CAN Filter 2 scale configuration [1] (in _CAN_FCR2, page 6)

Definition at line 6263 of file STM8AF_STM8S.h.

5.1.2.304 _CAN_FSC3

```
#define _CAN_FSC3 ((uint8_t) (0x03 << 5))
```

CAN Filter 3 scale configuration [1:0] (in _CAN_FCR2, page 6)

Definition at line 6266 of file STM8AF_STM8S.h.

5.1.2.305 _CAN_FSC30

```
#define _CAN_FSC30 ((uint8_t) (0x01 << 5))
```

CAN Filter 3 scale configuration [0] (in _CAN_FCR2, page 6)

Definition at line 6267 of file STM8AF_STM8S.h.

5.1.2.306 _CAN_FSC31

```
#define _CAN_FSC31 ((uint8_t) (0x01 << 6))
```

CAN Filter 3 scale configuration [1] (in _CAN_FCR2, page 6)

Definition at line 6268 of file STM8AF_STM8S.h.

5.1.2.307 _CAN_FSC4

```
#define _CAN_FSC4 ((uint8_t) (0x03 << 1))
```

CAN Filter 4 scale configuration [1:0] (in _CAN_FCR3, page 6)

Definition at line 6273 of file STM8AF_STM8S.h.

5.1.2.308 _CAN_FSC40

```
#define _CAN_FSC40 ((uint8_t) (0x01 << 1))
```

CAN Filter 4 scale configuration [0] (in _CAN_FCR3, page 6)

Definition at line 6274 of file STM8AF_STM8S.h.

5.1.2.309 _CAN_FSC41

```
#define _CAN_FSC41 ((uint8_t) (0x01 << 2))
```

CAN Filter 4 scale configuration [1] (in _CAN_FCR3, page 6)

Definition at line 6275 of file STM8AF_STM8S.h.

5.1.2.310 _CAN_FSC5

```
#define _CAN_FSC5 ((uint8_t) (0x03 << 5))
```

CAN Filter 5 scale configuration [1:0] (in _CAN_FCR3, page 6)

Definition at line 6278 of file STM8AF_STM8S.h.

5.1.2.311 _CAN_FSC50

```
#define _CAN_FSC50 ((uint8_t) (0x01 << 5))
```

CAN Filter 5 scale configuration [0] (in _CAN_FCR3, page 6)

Definition at line 6279 of file STM8AF_STM8S.h.

5.1.2.312 _CAN_FSC51

```
#define _CAN_FSC51 ((uint8_t) (0x01 << 6))
```

CAN Filter 5 scale configuration [1] (in _CAN_FCR3, page 6)

Definition at line 6280 of file STM8AF_STM8S.h.

5.1.2.313 _CAN_FULL

```
#define _CAN_FULL ((uint8_t) (0x01 << 3))
```

CAN FIFO Full [0] (in _CAN_RFR)

Definition at line 6133 of file STM8AF_STM8S.h.

5.1.2.314 _CAN_IDE

```
#define _CAN_IDE ((uint8_t) (0x01 << 6))
```

CAN Extended identifier [0] (in _CAN_MIDR1, page 0,1,5)

Definition at line 6182 of file STM8AF_STM8S.h.

5.1.2.315 _CAN_IER

```
#define _CAN_IER _SFR(uint8_t, CAN_AddressBase+0x05)
```

CAN interrupt enable register.

Definition at line 5952 of file STM8AF_STM8S.h.

5.1.2.316 _CAN_IER_RESET_VALUE

```
#define _CAN_IER_RESET_VALUE ((uint8_t) 0x00)
```

CAN interrupt enable register reset value.

Definition at line 6071 of file STM8AF_STM8S.h.

5.1.2.317 _CAN_INAK

```
#define _CAN_INAK ((uint8_t) (0x01 << 0))
```

CAN Initialization Acknowledge [0] (in _CAN_MSR)

Definition at line 6099 of file STM8AF_STM8S.h.

5.1.2.318 _CAN_INRQ

```
#define _CAN_INRQ ((uint8_t) (0x01 << 0))
```

CAN Channel Initialization Request [0] (in _CAN_MCR)

Definition at line 6089 of file STM8AF_STM8S.h.

5.1.2.319 _CAN_LBKM

```
#define _CAN_LBKM ((uint8_t) (0x01 << 0))
```

CAN Loop back mode [0] (in _CAN_DGR)

Definition at line 6147 of file STM8AF_STM8S.h.

5.1.2.320 _CAN_LEC

```
#define _CAN_LEC ((uint8_t) (0x07 << 4))
```

CAN Last error code [2:0] (in _CAN_ESR, page 6)

Definition at line 6190 of file STM8AF_STM8S.h.

5.1.2.321 _CAN_LEC0

```
#define _CAN_LEC0 ((uint8_t) (0x01 << 4))
```

CAN Last error code [0] (in _CAN_ESR, page 6)

Definition at line 6191 of file STM8AF_STM8S.h.

5.1.2.322 _CAN_LEC1

```
#define _CAN_LEC1 ((uint8_t) (0x01 << 5))
```

CAN Last error code [1] (in _CAN_ESR, page 6)

Definition at line 6192 of file STM8AF_STM8S.h.

5.1.2.323 _CAN_LEC2

```
#define _CAN_LEC2 ((uint8_t) (0x01 << 6))
```

CAN Last error code [3] (in _CAN_ESR, page 6)

Definition at line 6193 of file STM8AF_STM8S.h.

5.1.2.324 _CAN_LECIE

```
#define _CAN_LECIE ((uint8_t) (0x01 << 4))
```

CAN Last error code interrupt enable [0] (in _CAN_EIER, page 6)

Definition at line 6201 of file STM8AF_STM8S.h.

5.1.2.325 _CAN_LOW0

```
#define _CAN_LOW0 ((uint8_t) (0x01 << 5))
```

CAN Lowest Priority Flag for Mailbox 0 [0] (in _CAN_TPR)

Definition at line 6124 of file STM8AF_STM8S.h.

5.1.2.326 _CAN_LOW1

```
#define _CAN_LOW1 ((uint8_t) (0x01 << 6))
```

CAN Lowest Priority Flag for Mailbox 1 [0] (in _CAN_TPR)

Definition at line 6125 of file STM8AF_STM8S.h.

5.1.2.327 _CAN_LOW2

```
#define _CAN_LOW2 ((uint8_t) (0x01 << 7))
```

CAN Lowest Priority Flag for Mailbox 2 [0] (in _CAN_TPR)

Definition at line 6126 of file STM8AF_STM8S.h.

5.1.2.328 _CAN_MCR

```
#define _CAN_MCR _SFR(uint8_t, CAN_AddressBase+0x00)
```

CAN master control register.

Definition at line 5947 of file STM8AF_STM8S.h.

5.1.2.329 _CAN_MCR_RESET_VALUE

```
#define _CAN_MCR_RESET_VALUE ((uint8_t) 0x02)
```

CAN master control register reset value.

Definition at line 6066 of file STM8AF_STM8S.h.

5.1.2.330 _CAN_MCSR

```
#define _CAN_MCSR _SFR(uint8_t, CAN_AddressBase+0x08+0x00)
```

CAN message control/status register (page 0,1,5)

Definition at line 5957 of file STM8AF_STM8S.h.

5.1.2.331 _CAN_MCSR_RESET_VALUE

```
#define _CAN_MCSR_RESET_VALUE ((uint8_t) 0x00)
```

CAN message control/status register (page 0,1,5) reset value.

Definition at line 6075 of file STM8AF_STM8S.h.

5.1.2.332 _CAN_MDAR1

```
#define _CAN_MDAR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x06)
```

CAN mailbox data register 1 (page 0,1,5,7) */.

Definition at line 5963 of file STM8AF_STM8S.h.

5.1.2.333 _CAN_MDAR2

```
#define _CAN_MDAR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x07)
```

CAN mailbox data register 2 (page 0,1,5,7) */.

Definition at line 5964 of file STM8AF_STM8S.h.

5.1.2.334 _CAN_MDAR3

```
#define _CAN_MDAR3 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)
```

CAN mailbox data register 3 (page 0,1,5,7) */.

Definition at line 5965 of file STM8AF_STM8S.h.

5.1.2.335 _CAN_MDAR4

```
#define _CAN_MDAR4 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)
```

CAN mailbox data register 4 (page 0,1,5,7) */.

Definition at line 5966 of file STM8AF_STM8S.h.

5.1.2.336 _CAN_MDAR5

```
#define _CAN_MDAR5 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)
```

CAN mailbox data register 5 (page 0,1,5,7) */.

Definition at line 5967 of file STM8AF_STM8S.h.

5.1.2.337 _CAN_MDAR6

```
#define _CAN_MDAR6 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)
```

CAN mailbox data register 6 (page 0,1,5,7) */.

Definition at line 5968 of file STM8AF_STM8S.h.

5.1.2.338 _CAN_MDAR7

```
#define _CAN_MDAR7 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)
```

CAN mailbox data register 7 (page 0,1,5,7) */.

Definition at line 5969 of file STM8AF_STM8S.h.

5.1.2.339 _CAN_MDAR8

```
#define _CAN_MDAR8 _SFR(uint8_t, CAN_AddressBase+0x08+0x0D)
```

CAN mailbox data register 8 (page 0,1,5,7) */.

Definition at line 5970 of file STM8AF_STM8S.h.

5.1.2.340 _CAN_MDLCR

```
#define _CAN_MDLCR _SFR(uint8_t, CAN_AddressBase+0x08+0x01)
```

CAN mailbox data length control register (page 0,1,5,7)

Definition at line 5958 of file STM8AF_STM8S.h.

5.1.2.341 _CAN_MDLCR_RESET_VALUE

```
#define _CAN_MDLCR_RESET_VALUE ((uint8_t) 0x00)
```

CAN mailbox data length control register (page 0,1,5,7) reset value.

Definition at line 6076 of file STM8AF_STM8S.h.

5.1.2.342 _CAN_MFMIR

```
#define _CAN_MFMIR _SFR(uint8_t, CAN_AddressBase+0x08+0x00)
```

CAN mailbox filter match index register (page 7)

Definition at line 6048 of file STM8AF_STM8S.h.

5.1.2.343 _CAN_MFMIR_RESET_VALUE

```
#define _CAN_MFMIR_RESET_VALUE ((uint8_t) 0x00)
```

CAN mailbox filter match index register reset value.

Definition at line 6086 of file STM8AF_STM8S.h.

5.1.2.344 _CAN_MIDR1

```
#define _CAN_MIDR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x02)
```

CAN mailbox identifier register 1 (page 0,1,5,7)

Definition at line 5959 of file STM8AF_STM8S.h.

5.1.2.345 _CAN_MIDR2

```
#define _CAN_MIDR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x03)
```

CAN mailbox identifier register 2 (page 0,1,5,7)

Definition at line 5960 of file STM8AF_STM8S.h.

5.1.2.346 _CAN_MIDR3

```
#define _CAN_MIDR3 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)
```

CAN mailbox identifier register 3 (page 0,1,5,7)

Definition at line 5961 of file STM8AF_STM8S.h.

5.1.2.347 _CAN_MIDR4

```
#define _CAN_MIDR4 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)
```

CAN mailbox identifier register 4 (page 0,1,5,7)

Definition at line 5962 of file STM8AF_STM8S.h.

5.1.2.348 _CAN_MSR

```
#define _CAN_MSR _SFR(uint8_t, CAN_AddressBase+0x01)
```

CAN master status register.

Definition at line 5948 of file STM8AF_STM8S.h.

5.1.2.349 _CAN_MSR_RESET_VALUE

```
#define _CAN_MSR_RESET_VALUE ((uint8_t) 0x02)
```

CAN master status register reset value.

Definition at line 6067 of file STM8AF_STM8S.h.

5.1.2.350 _CAN_MTSRH

```
#define _CAN_MTSRH _SFR(uint8_t, CAN_AddressBase+0x08+0x0F)
```

CAN mailbox time stamp register high byte (page 0,1,5,7) */.

Definition at line 5972 of file STM8AF_STM8S.h.

5.1.2.351 _CAN_MTSRL

```
#define _CAN_MTSRL _SFR(uint8_t, CAN_AddressBase+0x08+0x0E)
```

CAN mailbox time stamp register low byte (page 0,1,5,7) */.

Definition at line 5971 of file STM8AF_STM8S.h.

5.1.2.352 _CAN_NART

```
#define _CAN_NART ((uint8_t) (0x01 << 4))
```

CAN Channel No Automatic Retransmission [0] (in _CAN_MCR)

Definition at line 6093 of file STM8AF_STM8S.h.

5.1.2.353 _CAN_PS

```
#define _CAN_PS ((uint8_t) (0x07 << 0))
```

CAN Page select [2:0] (in _CAN_PSR)

Definition at line 6155 of file STM8AF_STM8S.h.

5.1.2.354 _CAN_PS0

```
#define _CAN_PS0 ((uint8_t) (0x01 << 0))
```

CAN Page select [0] (in _CAN_PSR)

Definition at line 6156 of file STM8AF_STM8S.h.

5.1.2.355 _CAN_PS1

```
#define _CAN_PS1 ((uint8_t) (0x01 << 1))
```

CAN Page select [1] (in _CAN_PSR)

Definition at line 6157 of file STM8AF_STM8S.h.

5.1.2.356 _CAN_PS2

```
#define _CAN_PS2 ((uint8_t) (0x01 << 2))
```

CAN Page select [2] (in _CAN_PSR)

Definition at line 6158 of file STM8AF_STM8S.h.

5.1.2.357 _CAN_PSR

```
#define _CAN_PSR _SFR(uint8_t, CAN_AddressBase+0x07)
```

CAN page selection for below paged registers.

Definition at line 5954 of file STM8AF_STM8S.h.

5.1.2.358 _CAN_PSR_RESET_VALUE

```
#define _CAN_PSR_RESET_VALUE ((uint8_t) 0x00)
```

CAN page selection reset value.

Definition at line 6073 of file STM8AF_STM8S.h.

5.1.2.359 _CAN_RECR

```
#define _CAN_RECR _SFR(uint8_t, CAN_AddressBase+0x08+0x03)
```

CAN receive error counter register (page 6)

Definition at line 6036 of file STM8AF_STM8S.h.

5.1.2.360 _CAN_RECR_RESET_VALUE

```
#define _CAN_RECR_RESET_VALUE ((uint8_t) 0x00)
```

CAN receive error counter register (page 6) reset value.

Definition at line 6080 of file STM8AF_STM8S.h.

5.1.2.361 _CAN_RFLM

```
#define _CAN_RFLM ((uint8_t) (0x01 << 3))
```

CAN Channel Receive FIFO Locked Mode [0] (in _CAN_MCR)

Definition at line 6092 of file STM8AF_STM8S.h.

5.1.2.362 _CAN_RFOM

```
#define _CAN_RFOM ((uint8_t) (0x01 << 5))
```

CAN Release FIFO Output Mailbox [0] (in _CAN_RFR)

Definition at line 6135 of file STM8AF_STM8S.h.

5.1.2.363 _CAN_RFR

```
#define _CAN_RFR _SFR(uint8_t, CAN_AddressBase+0x04)
```

CAN receive FIFO register.

Definition at line 5951 of file STM8AF_STM8S.h.

5.1.2.364 _CAN_RFR_RESET_VALUE

```
#define _CAN_RFR_RESET_VALUE ((uint8_t) 0x00)
```

CAN receive FIFO register reset value.

Definition at line 6070 of file STM8AF_STM8S.h.

5.1.2.365 _CAN_RQCP

```
#define _CAN_RQCP ((uint8_t) (0x01 << 2))
```

CAN Request completed [0] (in _CAN_MCSR, page 0,1,5)

Definition at line 6164 of file STM8AF_STM8S.h.

5.1.2.366 _CAN_RQCP0

```
#define _CAN_RQCP0 ((uint8_t) (0x01 << 0))
```

CAN Request Completed for Mailbox 0 [0] (in _CAN_TSR)

Definition at line 6108 of file STM8AF_STM8S.h.

5.1.2.367 _CAN_RQCP1

```
#define _CAN_RQCP1 ((uint8_t) (0x01 << 1))
```

CAN Request Completed for Mailbox 1 [0] (in _CAN_TSR)

Definition at line 6109 of file STM8AF_STM8S.h.

5.1.2.368 _CAN_RQCP2

```
#define _CAN_RQCP2 ((uint8_t) (0x01 << 2))
```

CAN Request Completed for Mailbox 2 [0] (in _CAN_TSR)

Definition at line 6110 of file STM8AF_STM8S.h.

5.1.2.369 _CAN_RTR

```
#define _CAN_RTR ((uint8_t) (0x01 << 5))
```

CAN Remote transmission request [0] (in _CAN_MIDR1, page 0,1,5)

Definition at line 6181 of file STM8AF_STM8S.h.

5.1.2.370 _CAN_RX

```
#define _CAN_RX ((uint8_t) (0x01 << 5))
```

CAN Receive [0] (in _CAN_MSR)

Definition at line 6104 of file STM8AF_STM8S.h.

5.1.2.371 _CAN_RXS

```
#define _CAN_RXS ((uint8_t) (0x01 << 3))
```

CAN Rx Signal (=pin status) [0] (in _CAN_DGR)

Definition at line 6150 of file STM8AF_STM8S.h.

5.1.2.372 _CAN_SAMP

```
#define _CAN_SAMP ((uint8_t) (0x01 << 2))
```

CAN Last sample point [0] (in _CAN_DGR)

Definition at line 6149 of file STM8AF_STM8S.h.

5.1.2.373 _CAN_SILM

```
#define _CAN_SILM ((uint8_t) (0x01 << 1))
```

CAN Silent mode [0] (in _CAN_DGR)

Definition at line 6148 of file STM8AF_STM8S.h.

5.1.2.374 _CAN_SJW

```
#define _CAN_SJW ((uint8_t) (0x03 << 6))
```

CAN Resynchronization jump width [1:0] (in _CAN_EIER, page 6)

Definition at line 6214 of file STM8AF_STM8S.h.

5.1.2.375 _CAN_SJW0

```
#define _CAN_SJW0 ((uint8_t) (0x01 << 6))
```

CAN Resynchronization jump width [0] (in _CAN_EIER, page 6)

Definition at line 6215 of file STM8AF_STM8S.h.

5.1.2.376 _CAN_SJW1

```
#define _CAN_SJW1 ((uint8_t) (0x01 << 7))
```

CAN Resynchronization jump width [1] (in _CAN_EIER, page 6)

Definition at line 6216 of file STM8AF_STM8S.h.

5.1.2.377 _CAN_SLAK

```
#define _CAN_SLAK ((uint8_t) (0x01 << 1))
```

CAN Sleep Acknowledge [0] (in _CAN_MSR)

Definition at line 6100 of file STM8AF_STM8S.h.

5.1.2.378 _CAN_SLEEP

```
#define _CAN_SLEEP ((uint8_t) (0x01 << 1))
```

CAN Channel Sleep Mode Request [0] (in _CAN_MCR)

Definition at line 6090 of file STM8AF_STM8S.h.

5.1.2.379 _CAN_TECR

```
#define _CAN_TECR _SFR(uint8_t, CAN_AddressBase+0x08+0x02)
```

CAN transmit error counter register (page 6)

Definition at line 6035 of file STM8AF_STM8S.h.

5.1.2.380 _CAN_TECR_RESET_VALUE

```
#define _CAN_TECR_RESET_VALUE ((uint8_t) 0x00)
```

CAN transmit error counter register (page 6) reset value.

Definition at line 6079 of file STM8AF_STM8S.h.

5.1.2.381 _CAN_TERR

```
#define _CAN_TERR ((uint8_t) (0x01 << 5))
```

CAN Transmission error [0] (in _CAN_MCSR, page 0,1,5)

Definition at line 6167 of file STM8AF_STM8S.h.

5.1.2.382 _CAN_TGT

```
#define _CAN_TGT ((uint8_t) (0x01 << 7))
```

CAN Transmit global time [0] (in _CAN_MDLCR, page 0,1,5,7)

Definition at line 6177 of file STM8AF_STM8S.h.

5.1.2.383 _CAN_TME0

```
#define _CAN_TME0 ((uint8_t) (0x01 << 2))
```

CAN Transmit Mailbox 0 Empty [0] (in _CAN_TPR)

Definition at line 6121 of file STM8AF_STM8S.h.

5.1.2.384 _CAN_TME1

```
#define _CAN_TME1 ((uint8_t) (0x01 << 3))
```

CAN Transmit Mailbox 1 Empty [0] (in _CAN_TPR)

Definition at line 6122 of file STM8AF_STM8S.h.

5.1.2.385 _CAN_TME2

```
#define _CAN_TME2 ((uint8_t) (0x01 << 4))
```

CAN Transmit Mailbox 2 Empty [0] (in _CAN_TPR)

Definition at line 6123 of file STM8AF_STM8S.h.

5.1.2.386 _CAN_TMEIE

```
#define _CAN_TMEIE ((uint8_t) (0x01 << 0))
```

CAN Transmit Mailbox Empty Interrupt Enable [0] (in _CAN_IER)

Definition at line 6139 of file STM8AF_STM8S.h.

5.1.2.387 _CAN_TPR

```
#define _CAN_TPR _SFR(uint8_t, CAN_AddressBase+0x03)
```

CAN transmit priority register.

Definition at line 5950 of file STM8AF_STM8S.h.

5.1.2.388 _CAN_TPR_RESET_VALUE

```
#define _CAN_TPR_RESET_VALUE ((uint8_t) 0x0C)
```

CAN transmit priority register reset value.

Definition at line 6069 of file STM8AF_STM8S.h.

5.1.2.389 _CAN_TSR

```
#define _CAN_TSR _SFR(uint8_t, CAN_AddressBase+0x02)
```

CAN transmit status register.

Definition at line 5949 of file STM8AF_STM8S.h.

5.1.2.390 _CAN_TSR_RESET_VALUE

```
#define _CAN_TSR_RESET_VALUE ((uint8_t) 0x00)
```

CAN transmit status register reset value.

Definition at line 6068 of file STM8AF_STM8S.h.

5.1.2.391 _CAN_TTCM

```
#define _CAN_TTCM ((uint8_t) (0x01 << 7))
```

CAN Channel Time Triggered Communication Mode [0] (in _CAN_MCR)

Definition at line 6096 of file STM8AF_STM8S.h.

5.1.2.392 _CAN_TX

```
#define _CAN_TX ((uint8_t) (0x01 << 4))
```

CAN Transmit [0] (in _CAN_MSR)

Definition at line 6103 of file STM8AF_STM8S.h.

5.1.2.393 _CAN_TXFP

```
#define _CAN_TXFP ((uint8_t) (0x01 << 2))
```

CAN Channel Transmit FIFO Priority [0] (in _CAN_MCR)

Definition at line 6091 of file STM8AF_STM8S.h.

5.1.2.394 _CAN_TXM2E

```
#define _CAN_TXM2E ((uint8_t) (0x01 << 4))
```

CAN TX Mailbox 2 enable [0] (in _CAN_DGR)

Definition at line 6151 of file STM8AF_STM8S.h.

5.1.2.395 _CAN_TXOK

```
#define _CAN_TXOK ((uint8_t) (0x01 << 3))
```

CAN Transmission OK [0] (in _CAN_MCSR, page 0,1,5)

Definition at line 6165 of file STM8AF_STM8S.h.

5.1.2.396 _CAN_TXOK0

```
#define _CAN_TXOK0 ((uint8_t) (0x01 << 4))
```

CAN Transmission ok for Mailbox 0 [0] (in _CAN_TSR)

Definition at line 6112 of file STM8AF_STM8S.h.

5.1.2.397 _CAN_TXOK1

```
#define _CAN_TXOK1 ((uint8_t) (0x01 << 5))
```

CAN Transmission ok for Mailbox 1 [0] (in _CAN_TSR)

Definition at line 6113 of file STM8AF_STM8S.h.

5.1.2.398 _CAN_TXOK2

```
#define _CAN_TXOK2 ((uint8_t) (0x01 << 6))
```

CAN Transmission ok for Mailbox 2 [0] (in _CAN_TSR)

Definition at line 6114 of file STM8AF_STM8S.h.

5.1.2.399 _CAN_TXRQ

```
#define _CAN_TXRQ ((uint8_t) (0x01 << 0))
```

CAN Transmission mailbox request [0] (in _CAN_MCSR, page 0,1,5)

Definition at line 6162 of file STM8AF_STM8S.h.

5.1.2.400 _CAN_WKUI

```
#define _CAN_WKUI ((uint8_t) (0x01 << 3))
```

CAN Wakeup Interrupt [0] (in _CAN_MSR)

Definition at line 6102 of file STM8AF_STM8S.h.

5.1.2.401 _CAN_WKUIE

```
#define _CAN_WKUIE ((uint8_t) (0x01 << 7))
```

CAN Wakeup Interrupt Enable [0] (in _CAN_IER)

Definition at line 6144 of file STM8AF_STM8S.h.

5.1.2.402 _CFG

```
#define _CFG _SFR(CFG_t, CFG_AddressBase)
```

CFG struct/bit access.

Definition at line 6305 of file STM8AF_STM8S.h.

5.1.2.403 _CFG_AL

```
#define _CFG_AL ((uint8_t) (0x01 << 1))
```

Activation level [0].

Definition at line 6313 of file STM8AF_STM8S.h.

5.1.2.404 _CFG_GCR

```
#define _CFG_GCR _SFR(uint8_t, CFG_AddressBase+0x00)
```

Global configuration register (CFG_GCR)

Definition at line 6306 of file STM8AF_STM8S.h.

5.1.2.405 _CFG_GCR_RESET_VALUE

```
#define _CFG_GCR_RESET_VALUE ((uint8_t)0x00)
```

Definition at line 6309 of file STM8AF_STM8S.h.

5.1.2.406 _CFG_SWD

```
#define _CFG_SWD ((uint8_t) (0x01 << 0))
```

SWIM disable [0].

Definition at line 6312 of file STM8AF_STM8S.h.

5.1.2.407 _CLK

```
#define _CLK _SFR(CLK_t, CLK_AddressBase)
```

Clock module struct/bit access.

Definition at line 865 of file STM8AF_STM8S.h.

5.1.2.408 _CLK_ADC

```
#define _CLK_ADC ((uint8_t) (0x01 << 3))
```

clock enable ADC [0] (in _CLK_PCKENR2)

Definition at line 962 of file STM8AF_STM8S.h.

5.1.2.409 _CLK_AUX

```
#define _CLK_AUX ((uint8_t) (0x01 << 1))
```

Auxiliary oscillator connected to master clock [0] (in _CLK_CSSR)

Definition at line 943 of file STM8AF_STM8S.h.

5.1.2.410 _CLK_AWU

```
#define _CLK_AWU ((uint8_t) (0x01 << 2))
```

clock enable AWU [0] (in _CLK_PCKENR2)

Definition at line 961 of file STM8AF_STM8S.h.

5.1.2.411 _CLK_CAN

```
#define _CLK_CAN ((uint8_t) (0x01 << 7))
```

clock enable CAN [0] (in _CLK_PCKENR2)

Definition at line 964 of file STM8AF_STM8S.h.

5.1.2.412 _CLK_CCOBSY

```
#define _CLK_CCOBSY ((uint8_t) (0x01 << 6))
```

Configurable clock output busy [0] (in _CLK_CCOR)

Definition at line 956 of file STM8AF_STM8S.h.

5.1.2.413 _CLK_CCOEN

```
#define _CLK_CCOEN ((uint8_t) (0x01 << 0))
```

Configurable clock output enable [0] (in _CLK_CCOR)

Definition at line 949 of file STM8AF_STM8S.h.

5.1.2.414 _CLK_CCOR

```
#define _CLK_CCOR _SFR(uint8_t, CLK_AddressBase+0x09)
```

Configurable clock output register.

Definition at line 875 of file STM8AF_STM8S.h.

5.1.2.415 _CLK_CCOR_RESET_VALUE

```
#define _CLK_CCOR_RESET_VALUE ((uint8_t) 0x00)
```

Configurable clock output register reset value.

Definition at line 891 of file STM8AF_STM8S.h.

5.1.2.416 _CLK_CCORDY

```
#define _CLK_CCORDY ((uint8_t) (0x01 << 5))
```

Configurable clock output ready [0] (in _CLK_CCOR)

Definition at line 955 of file STM8AF_STM8S.h.

5.1.2.417 _CLK_CCOSEL

```
#define _CLK_CCOSEL ((uint8_t) (0x0F << 1))
```

Configurable clock output selection [3:0] (in _CLK_CCOR)

Definition at line 950 of file STM8AF_STM8S.h.

5.1.2.418 _CLK_CCOSEL0

```
#define _CLK_CCOSEL0 ((uint8_t) (0x01 << 1))
```

Configurable clock output selection [0] (in _CLK_CCOR)

Definition at line 951 of file STM8AF_STM8S.h.

5.1.2.419 _CLK_CCOSEL1

```
#define _CLK_CCOSEL1 ((uint8_t) (0x01 << 2))
```

Configurable clock output selection [1] (in _CLK_CCOR)

Definition at line 952 of file STM8AF_STM8S.h.

5.1.2.420 _CLK_CCOSEL2

```
#define _CLK_CCOSEL2 ((uint8_t) (0x01 << 3))
```

Configurable clock output selection [2] (in _CLK_CCOR)

Definition at line 953 of file STM8AF_STM8S.h.

5.1.2.421 _CLK_CCOSEL3

```
#define _CLK_CCOSEL3 ((uint8_t) (0x01 << 4))
```

Configurable clock output selection [3] (in _CLK_CCOR)

Definition at line 954 of file STM8AF_STM8S.h.

5.1.2.422 _CLK_CKDIVR

```
#define _CLK_CKDIVR _SFR(uint8_t, CLK_AddressBase+0x06)
```

Clock divider register.

Definition at line 872 of file STM8AF_STM8S.h.

5.1.2.423 _CLK_CKDIVR_RESET_VALUE

```
#define _CLK_CKDIVR_RESET_VALUE ((uint8_t) 0x18)
```

Clock divider register reset value.

Definition at line 887 of file STM8AF_STM8S.h.

5.1.2.424 _CLK_CMSR

```
#define _CLK_CMSR _SFR(uint8_t, CLK_AddressBase+0x03)
```

Clock master status register.

Definition at line 869 of file STM8AF_STM8S.h.

5.1.2.425 _CLK_CMSR_RESET_VALUE

```
#define _CLK_CMSR_RESET_VALUE ((uint8_t) 0xE1)
```

Clock master status reset value.

Definition at line 884 of file STM8AF_STM8S.h.

5.1.2.426 _CLK_CPUDIV

```
#define _CLK_CPUDIV ((uint8_t) (0x07 << 0))
```

CPU clock prescaler [2:0] (in _CLK_CKDIVR)

Definition at line 922 of file STM8AF_STM8S.h.

5.1.2.427 _CLK_CPUDIV0

```
#define _CLK_CPUDIV0 ((uint8_t) (0x01 << 0))
```

CPU clock prescaler [0] (in _CLK_CKDIVR)

Definition at line 923 of file STM8AF_STM8S.h.

5.1.2.428 _CLK_CPUDIV1

```
#define _CLK_CPUDIV1 ((uint8_t) (0x01 << 1))
```

CPU clock prescaler [1] (in _CLK_CKDIVR)

Definition at line 924 of file STM8AF_STM8S.h.

5.1.2.429 _CLK_CPUDIV2

```
#define _CLK_CPUDIV2 ((uint8_t) (0x01 << 2))
```

CPU clock prescaler [2] (in _CLK_CKDIVR)

Definition at line 925 of file STM8AF_STM8S.h.

5.1.2.430 _CLK_CSSD

```
#define _CLK_CSSD ((uint8_t) (0x01 << 3))
```

Clock security system detection [0] (in _CLK_CSSR)

Definition at line 945 of file STM8AF_STM8S.h.

5.1.2.431 _CLK_CSSDIE

```
#define _CLK_CSSDIE ((uint8_t) (0x01 << 2))
```

Clock security system detection interrupt enable [0] (in _CLK_CSSR)

Definition at line 944 of file STM8AF_STM8S.h.

5.1.2.432 _CLK_CSSEN

```
#define _CLK_CSSEN ((uint8_t) (0x01 << 0))
```

Clock security system enable [0] (in _CLK_CSSR)

Definition at line 942 of file STM8AF_STM8S.h.

5.1.2.433 _CLK_CSSR

```
#define _CLK_CSSR _SFR(uint8_t, CLK_AddressBase+0x08)
```

Clock security system register.

Definition at line 874 of file STM8AF_STM8S.h.

5.1.2.434 _CLK_CSSR_RESET_VALUE

```
#define _CLK_CSSR_RESET_VALUE ((uint8_t) 0x00)
```

Clock security system register reset value.

Definition at line 890 of file STM8AF_STM8S.h.

5.1.2.435 _CLK_ECKR

```
#define _CLK_ECKR _SFR(uint8_t, CLK_AddressBase+0x01)
```

External clock register.

Definition at line 867 of file STM8AF_STM8S.h.

5.1.2.436 _CLK_ECKR_HSERDY

```
#define _CLK_ECKR_HSERDY ((uint8_t) (0x01 << 1))
```

High speed external crystal oscillator ready [0] (in _CLK_ECKR)

Definition at line 906 of file STM8AF_STM8S.h.

5.1.2.437 _CLK_ECKR_RESET_VALUE

```
#define _CLK_ECKR_RESET_VALUE ((uint8_t) 0x00)
```

External clock register reset value.

Definition at line 883 of file STM8AF_STM8S.h.

5.1.2.438 _CLK_FHWU

```
#define _CLK_FHWU ((uint8_t) (0x01 << 2))
```

Fast wakeup from Halt/Active-halt modes [0] (in _CLK_ICKR)

Definition at line 898 of file STM8AF_STM8S.h.

5.1.2.439 _CLK_HSEEN

```
#define _CLK_HSEEN ((uint8_t) (0x01 << 0))
```

High speed external crystal oscillator enable [0] (in _CLK_ECKR)

Definition at line 905 of file STM8AF_STM8S.h.

5.1.2.440 _CLK_HSIDIV

```
#define _CLK_HSIDIV ((uint8_t) (0x03 << 3))
```

High speed internal clock prescaler [1:0] (in _CLK_CKDIVR)

Definition at line 926 of file STM8AF_STM8S.h.

5.1.2.441 _CLK_HSIDIV0

```
#define _CLK_HSIDIV0 ((uint8_t) (0x01 << 3))
```

High speed internal clock prescaler [0] (in _CLK_CKDIVR)

Definition at line 927 of file STM8AF_STM8S.h.

5.1.2.442 _CLK_HSIDIV1

```
#define _CLK_HSIDIV1 ((uint8_t) (0x01 << 4))
```

High speed internal clock prescaler [1] (in _CLK_CKDIVR)

Definition at line 928 of file STM8AF_STM8S.h.

5.1.2.443 _CLK_HSIEN

```
#define _CLK_HSIEN ((uint8_t) (0x01 << 0))
```

High speed internal RC oscillator enable [0] (in _CLK_ICKR)

Definition at line 896 of file STM8AF_STM8S.h.

5.1.2.444 _CLK_HSIRDY

```
#define _CLK_HSIRDY ((uint8_t) (0x01 << 1))
```

High speed internal oscillator ready [0] (in _CLK_ICKR)

Definition at line 897 of file STM8AF_STM8S.h.

5.1.2.445 _CLK_HSITRIM

```
#define _CLK_HSITRIM ((uint8_t) (0x0F << 0))
```

HSI trimming value (some devices only support 3 bits, see DS!) [3:0] (in _CLK_HSITRIMR)

Definition at line 967 of file STM8AF_STM8S.h.

5.1.2.446 _CLK_HSITRIM0

```
#define _CLK_HSITRIM0 ((uint8_t) (0x01 << 0))
```

HSI trimming value [0] (in _CLK_HSITRIMR)

Definition at line 968 of file STM8AF_STM8S.h.

5.1.2.447 _CLK_HSITRIM1

```
#define _CLK_HSITRIM1 ((uint8_t) (0x01 << 1))
```

HSI trimming value [1] (in _CLK_HSITRIMR)

Definition at line 969 of file STM8AF_STM8S.h.

5.1.2.448 _CLK_HSITRIM2

```
#define _CLK_HSITRIM2 ((uint8_t) (0x01 << 2))
```

HSI trimming value [2] (in _CLK_HSITRIMR)

Definition at line 970 of file STM8AF_STM8S.h.

5.1.2.449 _CLK_HSITRIM3

```
#define _CLK_HSITRIM3 ((uint8_t) (0x01 << 3))
```

HSI trimming value [3] (in _CLK_HSITRIMR)

Definition at line 971 of file STM8AF_STM8S.h.

5.1.2.450 _CLK_HSITRIMR

```
#define _CLK_HSITRIMR _SFR(uint8_t, CLK_AddressBase+0x0C)
```

HSI clock calibration trimming register.

Definition at line 878 of file STM8AF_STM8S.h.

5.1.2.451 _CLK_HSITRIMR_RESET_VALUE

```
#define _CLK_HSITRIMR_RESET_VALUE ((uint8_t) 0x00)
```

HSI clock calibration trimming register reset value.

Definition at line 892 of file STM8AF_STM8S.h.

5.1.2.452 _CLK_I2C

```
#define _CLK_I2C ((uint8_t) (0x01 << 0))
```

clock enable I2C [0] (in _CLK_PCKENR1)

Definition at line 932 of file STM8AF_STM8S.h.

5.1.2.453 _CLK_ICKR

```
#define _CLK_ICKR _SFR(uint8_t, CLK_AddressBase+0x00)
```

Internal clock register.

Definition at line 866 of file STM8AF_STM8S.h.

5.1.2.454 _CLK_ICKR_RESET_VALUE

```
#define _CLK_ICKR_RESET_VALUE ((uint8_t) 0x01)
```

Internal clock register reset value.

Definition at line 882 of file STM8AF_STM8S.h.

5.1.2.455 _CLK_LSIEN

```
#define _CLK_LSIEN ((uint8_t) (0x01 << 3))
```

Low speed internal RC oscillator enable [0] (in _CLK_ICKR)

Definition at line 899 of file STM8AF_STM8S.h.

5.1.2.456 _CLK_LSIRDY

```
#define _CLK_LSIRDY ((uint8_t) (0x01 << 4))
```

Low speed internal oscillator ready [0] (in _CLK_ICKR)

Definition at line 900 of file STM8AF_STM8S.h.

5.1.2.457 _CLK_PCKENR1

```
#define _CLK_PCKENR1 _SFR(uint8_t, CLK_AddressBase+0x07)
```

Peripheral clock gating register 1.

Definition at line 873 of file STM8AF_STM8S.h.

5.1.2.458 _CLK_PCKENR1_RESET_VALUE

```
#define _CLK_PCKENR1_RESET_VALUE ((uint8_t) 0xFF)
```

Peripheral clock gating register 1 reset value.

Definition at line 888 of file STM8AF_STM8S.h.

5.1.2.459 _CLK_PCKENR2

```
#define _CLK_PCKENR2 _SFR(uint8_t, CLK_AddressBase+0x0A)
```

Peripheral clock gating register 2.

Definition at line 876 of file STM8AF_STM8S.h.

5.1.2.460 _CLK_PCKENR2_RESET_VALUE

```
#define _CLK_PCKENR2_RESET_VALUE ((uint8_t) 0xFF)
```

Peripheral clock gating register 2 reset value.

Definition at line 889 of file STM8AF_STM8S.h.

5.1.2.461 _CLK_REGAH

```
#define _CLK_REGAH ((uint8_t) (0x01 << 5))
```

Regulator power off in Active-halt mode [0] (in _CLK_ICKR)

Definition at line 901 of file STM8AF_STM8S.h.

5.1.2.462 _CLK_SPI

```
#define _CLK_SPI ((uint8_t) (0x01 << 1))
```

clock enable SPI [0] (in _CLK_PCKENR1)

Definition at line 933 of file STM8AF_STM8S.h.

5.1.2.463 _CLK_SWBSY

```
#define _CLK_SWBSY ((uint8_t) (0x01 << 0))
```

Switch busy flag [0] (in _CLK_SWCR)

Definition at line 915 of file STM8AF_STM8S.h.

5.1.2.464 _CLK_SWCR

```
#define _CLK_SWCR _SFR(uint8_t, CLK_AddressBase+0x05)
```

Clock switch control register.

Definition at line 871 of file STM8AF_STM8S.h.

5.1.2.465 _CLK_SWCR_RESET_VALUE

```
#define _CLK_SWCR_RESET_VALUE ((uint8_t) 0x00)
```

Clock switch control reset value.

Definition at line 886 of file STM8AF_STM8S.h.

5.1.2.466 _CLK_SWEN

```
#define _CLK_SWEN ((uint8_t) (0x01 << 1))
```

Switch start/stop enable [0] (in _CLK_SWCR)

Definition at line 916 of file STM8AF_STM8S.h.

5.1.2.467 _CLK_SWI_HSE

```
#define _CLK_SWI_HSE ((uint8_t) 0xB4)
```

write to CLK_SWR for HSE clock (in _CLK_SWR)

Definition at line 912 of file STM8AF_STM8S.h.

5.1.2.468 _CLK_SWI_HSI

```
#define _CLK_SWI_HSI ((uint8_t) 0xE1)
```

write to CLK_SWR for HSI clock (in _CLK_SWR)

Definition at line 910 of file STM8AF_STM8S.h.

5.1.2.469 _CLK_SWI_LSI

```
#define _CLK_SWI_LSI ((uint8_t) 0xD2)
```

write to CLK_SWR for LSI clock (in _CLK_SWR)

Definition at line 911 of file STM8AF_STM8S.h.

5.1.2.470 _CLK_SWIEN

```
#define _CLK_SWIEN ((uint8_t) (0x01 << 2))
```

Clock switch interrupt enable [0] (in _CLK_SWCR)

Definition at line 917 of file STM8AF_STM8S.h.

5.1.2.471 _CLK_SWIF

```
#define _CLK_SWIF ((uint8_t) (0x01 << 3))
```

Clock switch interrupt flag [0] (in _CLK_SWCR)

Definition at line 918 of file STM8AF_STM8S.h.

5.1.2.472 _CLK_SWIMCCR

```
#define _CLK_SWIMCCR _SFR(uint8_t, CLK_AddressBase+0x0D)
```

SWIM clock control register.

Definition at line 879 of file STM8AF_STM8S.h.

5.1.2.473 _CLK_SWIMCCR_RESET_VALUE

```
#define _CLK_SWIMCCR_RESET_VALUE ((uint8_t) 0x00)
```

SWIM clock control register reset value.

Definition at line 893 of file STM8AF_STM8S.h.

5.1.2.474 _CLK_SWIMCLK

```
#define _CLK_SWIMCLK ((uint8_t) (0x01 << 0))
```

SWIM clock divider [0] (in _CLK_SWIMCCR)

Definition at line 975 of file STM8AF_STM8S.h.

5.1.2.475 _CLK_SWR

```
#define _CLK_SWR _SFR(uint8_t, CLK_AddressBase+0x04)
```

Clock master switch register.

Definition at line 870 of file STM8AF_STM8S.h.

5.1.2.476 _CLK_SWR_RESET_VALUE

```
#define _CLK_SWR_RESET_VALUE ((uint8_t) 0xE1)
```

Clock master switch reset value.

Definition at line 885 of file STM8AF_STM8S.h.

5.1.2.477 _CLK_TIM1

```
#define _CLK_TIM1 ((uint8_t) (0x01 << 7))
```

clock enable TIM1 [0] (in _CLK_PCKENR1)

Definition at line 939 of file STM8AF_STM8S.h.

5.1.2.478 _CLK_TIM2_TIM5

```
#define _CLK_TIM2_TIM5 ((uint8_t) (0x01 << 5))
```

clock enable TIM2/TIM5 [0] (in _CLK_PCKENR1)

Definition at line 937 of file STM8AF_STM8S.h.

5.1.2.479 _CLK_TIM3

```
#define _CLK_TIM3 ((uint8_t) (0x01 << 6))
```

clock enable TIM3 [0] (in _CLK_PCKENR1)

Definition at line 938 of file STM8AF_STM8S.h.

5.1.2.480 _CLK_TIM4_TIM6

```
#define _CLK_TIM4_TIM6 ((uint8_t) (0x01 << 4))
```

clock enable TIM4/TIM6 [0] (in _CLK_PCKENR1)

Definition at line 936 of file STM8AF_STM8S.h.

5.1.2.481 _CLK_UART1

```
#define _CLK_UART1 ((uint8_t) (0x01 << 2))
```

clock enable UART1 [0] (in _CLK_PCKENR1)

Definition at line 934 of file STM8AF_STM8S.h.

5.1.2.482 _CLK_UART2

```
#define _CLK_UART2 ((uint8_t) (0x01 << 3))
```

clock enable UART2 [0] (in _CLK_PCKENR1)

Definition at line 935 of file STM8AF_STM8S.h.

5.1.2.483 _EXTI

```
#define _EXTI _SFR(EXTI_t, EXTI_AddressBase)
```

External interrupt struct/bit access.

Definition at line 671 of file STM8AF_STM8S.h.

5.1.2.484 _EXTI_CR1

```
#define _EXTI_CR1 _SFR(uint8_t, EXTI_AddressBase+0x00)
```

External interrupt control register 1 (EXTI_CR1)

Definition at line 672 of file STM8AF_STM8S.h.

5.1.2.485 _EXTI_CR1_RESET_VALUE

```
#define _EXTI_CR1_RESET_VALUE ((uint8_t) 0x00)
```

External interrupt control register 1 reset value.

Definition at line 676 of file STM8AF_STM8S.h.

5.1.2.486 _EXTI_CR2

```
#define _EXTI_CR2 _SFR(uint8_t, EXTI_AddressBase+0x01)
```

External interrupt control register 2 (EXTI_CR2)

Definition at line 673 of file STM8AF_STM8S.h.

5.1.2.487 _EXTI_CR2_RESET_VALUE

```
#define _EXTI_CR2_RESET_VALUE ((uint8_t) 0x00)
```

External interrupt control register 2 reset value.

Definition at line 677 of file STM8AF_STM8S.h.

5.1.2.488 _EXTI_PAIS

```
#define _EXTI_PAIS ((uint8_t) (0x03 << 0))
```

External interrupt sensitivity for Port A [1:0] (in _EXTI_CR1)

Definition at line 680 of file STM8AF_STM8S.h.

5.1.2.489 _EXTI_PAIS0

```
#define _EXTI_PAIS0 ((uint8_t) (0x01 << 0))
```

External interrupt sensitivity for Port A [0] (in _EXTI_CR1)

Definition at line 681 of file STM8AF_STM8S.h.

5.1.2.490 _EXTI_PAIS1

```
#define _EXTI_PAIS1 ((uint8_t) (0x01 << 1))
```

External interrupt sensitivity for Port A [1] (in _EXTI_CR1)

Definition at line 682 of file STM8AF_STM8S.h.

5.1.2.491 _EXTI_PBIS

```
#define _EXTI_PBIS ((uint8_t) (0x03 << 2))
```

External interrupt sensitivity for Port B [1:0] (in _EXTI_CR1)

Definition at line 683 of file STM8AF_STM8S.h.

5.1.2.492 _EXTI_PBIS0

```
#define _EXTI_PBIS0 ((uint8_t) (0x01 << 2))
```

External interrupt sensitivity for Port B [0] (in _EXTI_CR1)

Definition at line 684 of file STM8AF_STM8S.h.

5.1.2.493 _EXTI_PBIS1

```
#define _EXTI_PBIS1 ((uint8_t) (0x01 << 3))
```

External interrupt sensitivity for Port B [1] (in _EXTI_CR1)

Definition at line 685 of file STM8AF_STM8S.h.

5.1.2.494 _EXTI_PCIS

```
#define _EXTI_PCIS ((uint8_t) (0x03 << 4))
```

External interrupt sensitivity for Port C [1:0] (in _EXTI_CR1)

Definition at line 686 of file STM8AF_STM8S.h.

5.1.2.495 _EXTI_PCIS0

```
#define _EXTI_PCIS0 ((uint8_t) (0x01 << 4))
```

External interrupt sensitivity for Port C [0] (in _EXTI_CR1)

Definition at line 687 of file STM8AF_STM8S.h.

5.1.2.496 _EXTI_PCIS1

```
#define _EXTI_PCIS1 ((uint8_t) (0x01 << 5))
```

External interrupt sensitivity for Port C [1] (in _EXTI_CR1)

Definition at line 688 of file STM8AF_STM8S.h.

5.1.2.497 _EXTI_PDIS

```
#define _EXTI_PDIS ((uint8_t) (0x03 << 6))
```

External interrupt sensitivity for Port D [1:0] (in _EXTI_CR1)

Definition at line 689 of file STM8AF_STM8S.h.

5.1.2.498 _EXTI_PDIS0

```
#define _EXTI_PDIS0 ((uint8_t) (0x01 << 6))
```

External interrupt sensitivity for Port D [0] (in _EXTI_CR1)

Definition at line 690 of file STM8AF_STM8S.h.

5.1.2.499 _EXTI_PDIS1

```
#define _EXTI_PDIS1 ((uint8_t) (0x01 << 7))
```

External interrupt sensitivity for Port D [1] (in _EXTI_CR1)

Definition at line 691 of file STM8AF_STM8S.h.

5.1.2.500 _EXTI_PEIS

```
#define _EXTI_PEIS ((uint8_t) (0x03 << 0))
```

Port E external interrupt sensitivity bits [1:0] (in _EXTI_CR2)

Definition at line 694 of file STM8AF_STM8S.h.

5.1.2.501 _EXTI_PEIS0

```
#define _EXTI_PEIS0 ((uint8_t) (0x01 << 0))
```

Port E external interrupt sensitivity bits [0] (in _EXTI_CR2)

Definition at line 695 of file STM8AF_STM8S.h.

5.1.2.502 _EXTI_PEIS1

```
#define _EXTI_PEIS1 ((uint8_t) (0x01 << 1))
```

Port E external interrupt sensitivity bits [1] (in _EXTI_CR2)

Definition at line 696 of file STM8AF_STM8S.h.

5.1.2.503 _EXTI_TLIS

```
#define _EXTI_TLIS ((uint8_t) (0x01 << 2))
```

Top level interrupt sensitivity [0] (in _EXTI_CR2)

Definition at line 697 of file STM8AF_STM8S.h.

5.1.2.504 _FLASH

```
#define _FLASH _SFR(FLASH_t, FLASH_AddressBase)
```

Flash struct/bit access.

Definition at line 586 of file STM8AF_STM8S.h.

5.1.2.505 _FLASH_AHALT

```
#define _FLASH_AHALT ((uint8_t) (0x01 << 2))
```

Power-down in Active-halt mode [0] (in _FLASH_CR1)

Definition at line 609 of file STM8AF_STM8S.h.

5.1.2.506 _FLASH_CR1

```
#define _FLASH_CR1 _SFR(uint8_t, FLASH_AddressBase+0x00)
```

Flash control register 1 (FLASH_CR1)

Definition at line 587 of file STM8AF_STM8S.h.

5.1.2.507 _FLASH_CR1_RESET_VALUE

```
#define _FLASH_CR1_RESET_VALUE ((uint8_t) 0x00)
```

Flash control register 1 reset value.

Definition at line 599 of file STM8AF_STM8S.h.

5.1.2.508 _FLASH_CR2

```
#define _FLASH_CR2 _SFR(uint8_t, FLASH_AddressBase+0x01)
```

Flash control register 2 (FLASH_CR2)

Definition at line 588 of file STM8AF_STM8S.h.

5.1.2.509 _FLASH_CR2_RESET_VALUE

```
#define _FLASH_CR2_RESET_VALUE ((uint8_t) 0x00)
```

Flash control register 2 reset value.

Definition at line 600 of file STM8AF_STM8S.h.

5.1.2.510 _FLASH_DUKR

```
#define _FLASH_DUKR _SFR(uint8_t, FLASH_AddressBase+0x0A)
```

Data EEPROM unprotection key register (FLASH_DUKR)

Definition at line 596 of file STM8AF_STM8S.h.

5.1.2.511 _FLASH_DUKR_RESET_VALUE

```
#define _FLASH_DUKR_RESET_VALUE ((uint8_t) 0x00)
```

Data EEPROM unprotection key reset value.

Definition at line 604 of file STM8AF_STM8S.h.

5.1.2.512 _FLASH_DUL

```
#define _FLASH_DUL ((uint8_t) (0x01 << 3))
```

Data EEPROM area unlocked flag [0] (in _FLASH_IAPSR)

Definition at line 635 of file STM8AF_STM8S.h.

5.1.2.513 _FLASH_EOP

```
#define _FLASH_EOP ((uint8_t) (0x01 << 2))
```

End of programming (write or erase operation) flag [0] (in _FLASH_IAPSR)

Definition at line 634 of file STM8AF_STM8S.h.

5.1.2.514 _FLASH_ERASE

```
#define _FLASH_ERASE ((uint8_t) (0x01 << 5))
```

Block erasing [0] (in _FLASH_CR2 and _FLASH_NCR2)

Definition at line 617 of file STM8AF_STM8S.h.

5.1.2.515 _FLASH_FIX

```
#define _FLASH_FIX ((uint8_t) (0x01 << 0))
```

Fixed Byte programming time [0] (in _FLASH_CR1)

Definition at line 607 of file STM8AF_STM8S.h.

5.1.2.516 _FLASH_FPR

```
#define _FLASH_FPR _SFR(uint8_t, FLASH_AddressBase+0x03)
```

Flash protection register (FLASH_FPR)

Definition at line 590 of file STM8AF_STM8S.h.

5.1.2.517 _FLASH_FPRG

```
#define _FLASH_FPRG ((uint8_t) (0x01 << 4))
```

Fast block programming [0] (in _FLASH_CR2 and _FLASH_NCR2)

Definition at line 616 of file STM8AF_STM8S.h.

5.1.2.518 _FLASH_HALT

```
#define _FLASH_HALT ((uint8_t) (0x01 << 3))
```

Power-down in Halt mode [0] (in _FLASH_CR1)

Definition at line 610 of file STM8AF_STM8S.h.

5.1.2.519 _FLASH_HVOFF

```
#define _FLASH_HVOFF ((uint8_t) (0x01 << 5))
```

End of high voltage flag [0] (in _FLASH_IAPSR)

Definition at line 637 of file STM8AF_STM8S.h.

5.1.2.520 _FLASH_IAPSR

```
#define _FLASH_IAPSR _SFR(uint8_t, FLASH_AddressBase+0x05)
```

Flash status register (FLASH_IAPSR)

Definition at line 592 of file STM8AF_STM8S.h.

5.1.2.521 _FLASH_IAPSR_RESET_VALUE

```
#define _FLASH_IAPSR_RESET_VALUE ((uint8_t) 0x40)
```

Flash status register reset value.

Definition at line 602 of file STM8AF_STM8S.h.

5.1.2.522 _FLASH_IE

```
#define _FLASH_IE ((uint8_t) (0x01 << 1))
```

Flash Interrupt enable [0] (in _FLASH_CR1)

Definition at line 608 of file STM8AF_STM8S.h.

5.1.2.523 _FLASH_NCR2

```
#define _FLASH_NCR2 _SFR(uint8_t, FLASH_AddressBase+0x02)
```

complementary Flash control register 2 (FLASH_NCR2)

Definition at line 589 of file STM8AF_STM8S.h.

5.1.2.524 _FLASH_NCR2_RESET_VALUE

```
#define _FLASH_NCR2_RESET_VALUE ((uint8_t) 0xFF)
```

complementary Flash control register 2 reset value

Definition at line 601 of file STM8AF_STM8S.h.

5.1.2.525 _FLASH_NFPR

```
#define _FLASH_NFPR _SFR(uint8_t, FLASH_AddressBase+0x04)
```

complementary Flash protection register (FLASH_NFPR)

Definition at line 591 of file STM8AF_STM8S.h.

5.1.2.526 _FLASH_OPT

```
#define _FLASH_OPT ((uint8_t) (0x01 << 7))
```

Write option bytes [0] (in _FLASH_CR2 and _FLASH_NCR2)

Definition at line 619 of file STM8AF_STM8S.h.

5.1.2.527 _FLASH_PRG

```
#define _FLASH_PRG ((uint8_t) (0x01 << 0))
```

Standard block programming [0] (in _FLASH_CR2 and _FLASH_NCR2)

Definition at line 614 of file STM8AF_STM8S.h.

5.1.2.528 _FLASH_PUKR

```
#define _FLASH_PUKR _SFR(uint8_t, FLASH_AddressBase+0x08)
```

Flash program memory unprotecting key register (FLASH_PUKR)

Definition at line 594 of file STM8AF_STM8S.h.

5.1.2.529 _FLASH_PUKR_RESET_VALUE

```
#define _FLASH_PUKR_RESET_VALUE ((uint8_t) 0x00)
```

Flash program memory unprotecting key reset value.

Definition at line 603 of file STM8AF_STM8S.h.

5.1.2.530 _FLASH_PUL

```
#define _FLASH_PUL ((uint8_t) (0x01 << 1))
```

Flash Program memory unlocked flag [0] (in _FLASH_IAPSR)

Definition at line 633 of file STM8AF_STM8S.h.

5.1.2.531 _FLASH_WPB

```
#define _FLASH_WPB ((uint8_t) (0x3F << 0))
```

User boot code area protection bits [5:0] (in _FLASH_FPR and _FLASH_NFPR)

Definition at line 622 of file STM8AF_STM8S.h.

5.1.2.532 _FLASH_WPB0

```
#define _FLASH_WPB0 ((uint8_t) (0x01 << 0))
```

User boot code area protection bit [0] (in _FLASH_FPR and _FLASH_NFPR)

Definition at line 623 of file STM8AF_STM8S.h.

5.1.2.533 _FLASH_WPB1

```
#define _FLASH_WPB1 ((uint8_t) (0x01 << 1))
```

User boot code area protection bit [1] (in _FLASH_FPR and _FLASH_NFPR)

Definition at line 624 of file STM8AF_STM8S.h.

5.1.2.534 _FLASH_WPB2

```
#define _FLASH_WPB2 ((uint8_t) (0x01 << 2))
```

User boot code area protection bit [2] (in _FLASH_FPR and _FLASH_NFPR)

Definition at line 625 of file STM8AF_STM8S.h.

5.1.2.535 _FLASH_WPB3

```
#define _FLASH_WPB3 ((uint8_t) (0x01 << 3))
```

User boot code area protection bit [3] (in _FLASH_FPR and _FLASH_NFPR)

Definition at line 626 of file STM8AF_STM8S.h.

5.1.2.536 _FLASH_WPB4

```
#define _FLASH_WPB4 ((uint8_t) (0x01 << 4))
```

User boot code area protection bit [4] (in _FLASH_FPR and _FLASH_NFPR)

Definition at line 627 of file STM8AF_STM8S.h.

5.1.2.537 _FLASH_WPB5

```
#define _FLASH_WPB5 ((uint8_t) (0x01 << 5))
```

User boot code area protection bit [5] (in _FLASH_FPR and _FLASH_NFPR)

Definition at line 628 of file STM8AF_STM8S.h.

5.1.2.538 _FLASH_WPRG

```
#define _FLASH_WPRG ((uint8_t) (0x01 << 6))
```

Word programming [0] (in _FLASH_CR2 and _FLASH_NCR2)

Definition at line 618 of file STM8AF_STM8S.h.

5.1.2.539 _FLASH_WR_PG_DIS

```
#define _FLASH_WR_PG_DIS ((uint8_t) (0x01 << 0))
```

Write attempted to protected page flag [0] (in _FLASH_IAPSR)

Definition at line 632 of file STM8AF_STM8S.h.

5.1.2.540 _GPIO_CR1_RESET_VALUE

```
#define _GPIO_CR1_RESET_VALUE ((uint8_t) 0x00)
```

port control register 1 reset value

Definition at line 481 of file STM8AF_STM8S.h.

5.1.2.541 _GPIO_CR2_RESET_VALUE

```
#define _GPIO_CR2_RESET_VALUE ((uint8_t) 0x00)
```

port control register 2 reset value

Definition at line 482 of file STM8AF_STM8S.h.

5.1.2.542 _GPIO_DDR_RESET_VALUE

```
#define _GPIO_DDR_RESET_VALUE ((uint8_t) 0x00)
```

port direction register reset value

Definition at line 480 of file STM8AF_STM8S.h.

5.1.2.543 _GPIO_ODR_RESET_VALUE

```
#define _GPIO_ODR_RESET_VALUE ((uint8_t) 0x00)
```

port output register reset value

Definition at line 479 of file STM8AF_STM8S.h.

5.1.2.544 _GPIO_PIN0

```
#define _GPIO_PIN0 ((uint8_t) (0x01 << 0))
```

port bit mask for pin 0 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 485 of file STM8AF_STM8S.h.

5.1.2.545 _GPIO_PIN1

```
#define _GPIO_PIN1 ((uint8_t) (0x01 << 1))
```

port bit mask for pin 1 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 486 of file STM8AF_STM8S.h.

5.1.2.546 _GPIO_PIN2

```
#define _GPIO_PIN2 ((uint8_t) (0x01 << 2))
```

port bit mask for pin 2 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 487 of file STM8AF_STM8S.h.

5.1.2.547 _GPIO_PIN3

```
#define _GPIO_PIN3 ((uint8_t) (0x01 << 3))
```

port bit mask for pin 3 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 488 of file STM8AF_STM8S.h.

5.1.2.548 _GPIO_PIN4

```
#define _GPIO_PIN4 ((uint8_t) (0x01 << 4))
```

port bit mask for pin 4 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 489 of file STM8AF_STM8S.h.

5.1.2.549 _GPIO_PIN5

```
#define _GPIO_PIN5 ((uint8_t) (0x01 << 5))
```

port bit mask for pin 5 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 490 of file STM8AF_STM8S.h.

5.1.2.550 _GPIO_PIN6

```
#define _GPIO_PIN6 ((uint8_t) (0x01 << 6))
```

port bit mask for pin 6 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 491 of file STM8AF_STM8S.h.

5.1.2.551 _GPIO_PIN7

```
#define _GPIO_PIN7 ((uint8_t) (0x01 << 7))
```

port bit mask for pin 7 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 492 of file STM8AF_STM8S.h.

5.1.2.552 _GPIOA

```
#define _GPIOA _SFR(PORT_t, PORTA_AddressBase)
```

port A struct/bit access

Definition at line 390 of file STM8AF_STM8S.h.

5.1.2.553 _GPIOA_CR1

```
#define _GPIOA_CR1 _SFR(uint8_t, PORTA_AddressBase+0x03)
```

port A control register 1

Definition at line 394 of file STM8AF_STM8S.h.

5.1.2.554 _GPIOA_CR2

```
#define _GPIOA_CR2 _SFR(uint8_t, PORTA_AddressBase+0x04)
```

port A control register 2

Definition at line 395 of file STM8AF_STM8S.h.

5.1.2.555 _GPIOA_DDR

```
#define _GPIOA_DDR _SFR(uint8_t, PORTA_AddressBase+0x02)
```

port A direction register

Definition at line 393 of file STM8AF_STM8S.h.

5.1.2.556 _GPIOA_IDR

```
#define _GPIOA_IDR _SFR(uint8_t, PORTA_AddressBase+0x01)
```

port A input register

Definition at line 392 of file STM8AF_STM8S.h.

5.1.2.557 _GPIOA_ODR

```
#define _GPIOA_ODR _SFR(uint8_t, PORTA_AddressBase+0x00)
```

port A output register

Definition at line 391 of file STM8AF_STM8S.h.

5.1.2.558 _GPIOB

```
#define _GPIOB _SFR(PORT_t, PORTB_AddressBase)
```

port B struct/bit access

Definition at line 400 of file STM8AF_STM8S.h.

5.1.2.559 _GPIOB_CR1

```
#define _GPIOB_CR1 _SFR(uint8_t, PORTB_AddressBase+0x03)
```

port B control register 1

Definition at line 404 of file STM8AF_STM8S.h.

5.1.2.560 _GPIOB_CR2

```
#define _GPIOB_CR2 _SFR(uint8_t, PORTB_AddressBase+0x04)
```

port B control register 2

Definition at line 405 of file STM8AF_STM8S.h.

5.1.2.561 _GPIOB_DDR

```
#define _GPIOB_DDR _SFR(uint8_t, PORTB_AddressBase+0x02)
```

port B direction register

Definition at line 403 of file STM8AF_STM8S.h.

5.1.2.562 _GPIOB_IDR

```
#define _GPIOB_IDR _SFR(uint8_t, PORTB_AddressBase+0x01)
```

port B input register

Definition at line 402 of file STM8AF_STM8S.h.

5.1.2.563 _GPIOB_ODR

```
#define _GPIOB_ODR _SFR(uint8_t, PORTB_AddressBase+0x00)
```

port B output register

Definition at line 401 of file STM8AF_STM8S.h.

5.1.2.564 _GPIOC

```
#define _GPIOC _SFR(PORT_t, PORTC_AddressBase)
```

port C struct/bit access

Definition at line 410 of file STM8AF_STM8S.h.

5.1.2.565 _GPIOC_CR1

```
#define _GPIOC_CR1 _SFR(uint8_t, PORTC_AddressBase+0x03)
```

port C control register 1

Definition at line 414 of file STM8AF_STM8S.h.

5.1.2.566 _GPIOC_CR2

```
#define _GPIOC_CR2 _SFR(uint8_t, PORTC_AddressBase+0x04)
```

port C control register 2

Definition at line 415 of file STM8AF_STM8S.h.

5.1.2.567 _GPIOC_DDR

```
#define _GPIOC_DDR _SFR(uint8_t, PORTC_AddressBase+0x02)
```

port C direction register

Definition at line 413 of file STM8AF_STM8S.h.

5.1.2.568 _GPIOC_IDR

```
#define _GPIOC_IDR _SFR(uint8_t, PORTC_AddressBase+0x01)
```

port C input register

Definition at line 412 of file STM8AF_STM8S.h.

5.1.2.569 _GPIOC_ODR

```
#define _GPIOC_ODR _SFR(uint8_t, PORTC_AddressBase+0x00)
```

port C output register

Definition at line 411 of file STM8AF_STM8S.h.

5.1.2.570 _GPIOD

```
#define _GPIOD _SFR(PORT_t, PORTD_AddressBase)
```

port D struct/bit access

Definition at line 420 of file STM8AF_STM8S.h.

5.1.2.571 _GPIOD_CR1

```
#define _GPIOD_CR1 _SFR(uint8_t, PORTD_AddressBase+0x03)
```

port D control register 1

Definition at line 424 of file STM8AF_STM8S.h.

5.1.2.572 _GPIOD_CR2

```
#define _GPIOD_CR2 _SFR(uint8_t, PORTD_AddressBase+0x04)
```

port D control register 2

Definition at line 425 of file STM8AF_STM8S.h.

5.1.2.573 _GPIOD_DDR

```
#define _GPIOD_DDR _SFR(uint8_t, PORTD_AddressBase+0x02)
```

port D direction register

Definition at line 423 of file STM8AF_STM8S.h.

5.1.2.574 _GPIOD_IDR

```
#define _GPIOD_IDR _SFR(uint8_t, PORTD_AddressBase+0x01)
```

port D input register

Definition at line 422 of file STM8AF_STM8S.h.

5.1.2.575 _GPIOD_ODR

```
#define _GPIOD_ODR _SFR(uint8_t, PORTD_AddressBase+0x00)
```

port D output register

Definition at line 421 of file STM8AF_STM8S.h.

5.1.2.576 _GPIOE

```
#define _GPIOE _SFR(PORT_t, PORTE_AddressBase)
```

port E struct/bit access

Definition at line 430 of file STM8AF_STM8S.h.

5.1.2.577 _GPIOE_CR1

```
#define _GPIOE_CR1 _SFR(uint8_t, PORTE_AddressBase+0x03)
```

port E control register 1

Definition at line 434 of file STM8AF_STM8S.h.

5.1.2.578 _GPIOE_CR2

```
#define _GPIOE_CR2 _SFR(uint8_t, PORTE_AddressBase+0x04)
```

port E control register 2

Definition at line 435 of file STM8AF_STM8S.h.

5.1.2.579 _GPIOE_DDR

```
#define _GPIOE_DDR _SFR(uint8_t, PORTE_AddressBase+0x02)
```

port E direction register

Definition at line 433 of file STM8AF_STM8S.h.

5.1.2.580 _GPIOE_IDR

```
#define _GPIOE_IDR _SFR(uint8_t, PORTE_AddressBase+0x01)
```

port E input register

Definition at line 432 of file STM8AF_STM8S.h.

5.1.2.581 _GPIOE_ODR

```
#define _GPIOE_ODR _SFR(uint8_t, PORTE_AddressBase+0x00)
```

port E output register

Definition at line 431 of file STM8AF_STM8S.h.

5.1.2.582 _GPIOF

```
#define _GPIOF _SFR(PORT_t, PORTF_AddressBase)
```

port F struct/bit access

Definition at line 440 of file STM8AF_STM8S.h.

5.1.2.583 _GPIOF_CR1

```
#define _GPIOF_CR1 _SFR(uint8_t, PORTF_AddressBase+0x03)
```

port F control register 1

Definition at line 444 of file STM8AF_STM8S.h.

5.1.2.584 _GPIOF_CR2

```
#define _GPIOF_CR2 _SFR(uint8_t, PORTF_AddressBase+0x04)
```

port F control register 2

Definition at line 445 of file STM8AF_STM8S.h.

5.1.2.585 _GPIOF_DDR

```
#define _GPIOF_DDR _SFR(uint8_t, PORTF_AddressBase+0x02)
```

port F direction register

Definition at line 443 of file STM8AF_STM8S.h.

5.1.2.586 _GPIOF_IDR

```
#define _GPIOF_IDR _SFR(uint8_t, PORTF_AddressBase+0x01)
```

port F input register

Definition at line 442 of file STM8AF_STM8S.h.

5.1.2.587 _GPIOF_ODR

```
#define _GPIOF_ODR _SFR(uint8_t, PORTF_AddressBase+0x00)
```

port F output register

Definition at line 441 of file STM8AF_STM8S.h.

5.1.2.588 _GPIOG

```
#define _GPIOG _SFR(PORT_t, PORTG_AddressBase)
```

port G struct/bit access

Definition at line 450 of file STM8AF_STM8S.h.

5.1.2.589 _GPIOG_CR1

```
#define _GPIOG_CR1 _SFR(uint8_t, PORTG_AddressBase+0x03)
```

port G control register 1

Definition at line 454 of file STM8AF_STM8S.h.

5.1.2.590 _GPIOG_CR2

```
#define _GPIOG_CR2 _SFR(uint8_t, PORTG_AddressBase+0x04)
```

port G control register 2

Definition at line 455 of file STM8AF_STM8S.h.

5.1.2.591 _GPIOG_DDR

```
#define _GPIOG_DDR _SFR(uint8_t, PORTG_AddressBase+0x02)
```

port G direction register

Definition at line 453 of file STM8AF_STM8S.h.

5.1.2.592 _GPIOG_IDR

```
#define _GPIOG_IDR _SFR(uint8_t, PORTG_AddressBase+0x01)
```

port G input register

Definition at line 452 of file STM8AF_STM8S.h.

5.1.2.593 _GPIOG_ODR

```
#define _GPIOG_ODR _SFR(uint8_t, PORTG_AddressBase+0x00)
```

port G output register

Definition at line 451 of file STM8AF_STM8S.h.

5.1.2.594 _GPIOH

```
#define _GPIOH _SFR(PORT_t, PORTH_AddressBase)
```

port H struct/bit access

Definition at line 460 of file STM8AF_STM8S.h.

5.1.2.595 _GPIOH_CR1

```
#define _GPIOH_CR1 _SFR(uint8_t, PORTH_AddressBase+0x03)
```

port H control register 1

Definition at line 464 of file STM8AF_STM8S.h.

5.1.2.596 _GPIOH_CR2

```
#define _GPIOH_CR2 _SFR(uint8_t, PORTH_AddressBase+0x04)
```

port H control register 2

Definition at line 465 of file STM8AF_STM8S.h.

5.1.2.597 _GPIOH_DDR

```
#define _GPIOH_DDR _SFR(uint8_t, PORTH_AddressBase+0x02)
```

port H direction register

Definition at line 463 of file STM8AF_STM8S.h.

5.1.2.598 _GPIOH_IDR

```
#define _GPIOH_IDR _SFR(uint8_t, PORTH_AddressBase+0x01)
```

port H input register

Definition at line 462 of file STM8AF_STM8S.h.

5.1.2.599 _GPIOH_ODR

```
#define _GPIOH_ODR _SFR(uint8_t, PORTH_AddressBase+0x00)
```

port H output register

Definition at line 461 of file STM8AF_STM8S.h.

5.1.2.600 _GPIOI

```
#define _GPIOI _SFR(PORT_t, PORTI_AddressBase)
```

port I struct/bit access

Definition at line 470 of file STM8AF_STM8S.h.

5.1.2.601 _GPIOI_CR1

```
#define _GPIOI_CR1 _SFR(uint8_t, PORTI_AddressBase+0x03)
```

port I control register 1

Definition at line 474 of file STM8AF_STM8S.h.

5.1.2.602 _GPIOI_CR2

```
#define _GPIOI_CR2 _SFR(uint8_t, PORTI_AddressBase+0x04)
```

port I control register 2

Definition at line 475 of file STM8AF_STM8S.h.

5.1.2.603 _GPIOI_DDR

```
#define _GPIOI_DDR _SFR(uint8_t, PORTI_AddressBase+0x02)
```

port I direction register

Definition at line 473 of file STM8AF_STM8S.h.

5.1.2.604 _GPIOI_IDR

```
#define _GPIOI_IDR _SFR(uint8_t, PORTI_AddressBase+0x01)
```

port I input register

Definition at line 472 of file STM8AF_STM8S.h.

5.1.2.605 _GPIOI_ODR

```
#define _GPIOI_ODR _SFR(uint8_t, PORTI_AddressBase+0x00)
```

port I output register

Definition at line 471 of file STM8AF_STM8S.h.

5.1.2.606 _I2C

```
#define _I2C _SFR(I2C_t, I2C_AddressBase)
```

register for SPI control

I2C struct/bit access

Definition at line 1485 of file STM8AF_STM8S.h.

5.1.2.607 _I2C_ACK

```
#define _I2C_ACK ((uint8_t) (0x01 << 2))
```

I2C Acknowledge enable [0] (in _I2C_CR2)

Definition at line 1526 of file STM8AF_STM8S.h.

5.1.2.608 _I2C_ADD0

```
#define _I2C_ADD0 ((uint8_t) (0x01 << 0))
```

I2C Interface address [0] (in 10-bit address mode) (in _I2C_OARL)

Definition at line 1542 of file STM8AF_STM8S.h.

5.1.2.609 _I2C_ADD1

```
#define _I2C_ADD1 ((uint8_t) (0x01 << 1))
```

I2C Interface address [1] (in _I2C_OARL)

Definition at line 1543 of file STM8AF_STM8S.h.

5.1.2.610 _I2C_ADD10

```
#define _I2C_ADD10 ((uint8_t) (0x01 << 3))
```

I2C 10-bit header sent (Master mode) [0] (in _I2C_SR1)

Definition at line 1564 of file STM8AF_STM8S.h.

5.1.2.611 _I2C_ADD2

```
#define _I2C_ADD2 ((uint8_t) (0x01 << 2))
```

I2C Interface address [2] (in _I2C_OARL)

Definition at line 1544 of file STM8AF_STM8S.h.

5.1.2.612 _I2C_ADD3

```
#define _I2C_ADD3 ((uint8_t) (0x01 << 3))
```

I2C Interface address [3] (in _I2C_OARL)

Definition at line 1545 of file STM8AF_STM8S.h.

5.1.2.613 _I2C_ADD4

```
#define _I2C_ADD4 ((uint8_t) (0x01 << 4))
```

I2C Interface address [4] (in _I2C_OARL)

Definition at line 1546 of file STM8AF_STM8S.h.

5.1.2.614 _I2C_ADD5

```
#define _I2C_ADD5 ((uint8_t) (0x01 << 5))
```

I2C Interface address [5] (in _I2C_OARL)

Definition at line 1547 of file STM8AF_STM8S.h.

5.1.2.615 _I2C_ADD6

```
#define _I2C_ADD6 ((uint8_t) (0x01 << 6))
```

I2C Interface address [6] (in _I2C_OARL)

Definition at line 1548 of file STM8AF_STM8S.h.

5.1.2.616 _I2C_ADD7

```
#define _I2C_ADD7 ((uint8_t) (0x01 << 7))
```

I2C Interface address [7] (in _I2C_OARL)

Definition at line 1549 of file STM8AF_STM8S.h.

5.1.2.617 _I2C_ADD8

```
#define _I2C_ADD8 ((uint8_t) (0x01 << 1))
```

I2C Interface address [8] (in _I2C_OARH)

Definition at line 1554 of file STM8AF_STM8S.h.

5.1.2.618 _I2C_ADD9

```
#define _I2C_ADD9 ((uint8_t) (0x01 << 2))
```

I2C Interface address [9] (in _I2C_OARH)

Definition at line 1555 of file STM8AF_STM8S.h.

5.1.2.619 _I2C_ADD_8_9

```
#define _I2C_ADD_8_9 ((uint8_t) (0x03 << 1))
```

I2C Interface address [9:8] (in 10-bit address mode) (in _I2C_OARH)

Definition at line 1553 of file STM8AF_STM8S.h.

5.1.2.620 _I2C_ADDCONF

```
#define _I2C_ADDCONF ((uint8_t) (0x01 << 6))
```

I2C Address mode configuration [0] (in _I2C_OARH)

Definition at line 1557 of file STM8AF_STM8S.h.

5.1.2.621 _I2C_ADDMODE

```
#define _I2C_ADDMODE ((uint8_t) (0x01 << 7))
```

I2C 7-/10-bit addressing mode (Slave mode) [0] (in _I2C_OARH)

Definition at line 1558 of file STM8AF_STM8S.h.

5.1.2.622 _I2C_ADDR

```
#define _I2C_ADDR ((uint8_t) (0x01 << 1))
```

I2C Address sent (Master mode) / matched (Slave mode) [0] (in _I2C_SR1)

Definition at line 1562 of file STM8AF_STM8S.h.

5.1.2.623 _I2C_AF

```
#define _I2C_AF ((uint8_t) (0x01 << 2))
```

I2C Acknowledge failure [0] (in _I2C_SR2)

Definition at line 1573 of file STM8AF_STM8S.h.

5.1.2.624 _I2C_ARLO

```
#define _I2C_ARLO ((uint8_t) (0x01 << 1))
```

I2C Arbitration lost (Master mode) [0] (in _I2C_SR2)

Definition at line 1572 of file STM8AF_STM8S.h.

5.1.2.625 _I2C_BERR

```
#define _I2C_BERR ((uint8_t) (0x01 << 0))
```

I2C Bus error [0] (in _I2C_SR2)

Definition at line 1571 of file STM8AF_STM8S.h.

5.1.2.626 _I2C_BTF

```
#define _I2C_BTF ((uint8_t) (0x01 << 2))
```

I2C Byte transfer finished [0] (in _I2C_SR1)

Definition at line 1563 of file STM8AF_STM8S.h.

5.1.2.627 _I2C_BUSY

```
#define _I2C_BUSY ((uint8_t) (0x01 << 1))
```

I2C Bus busy [0] (in _I2C_SR3)

Definition at line 1581 of file STM8AF_STM8S.h.

5.1.2.628 _I2C_CCR

```
#define _I2C_CCR ((uint8_t) (0x0F << 0))
```

I2C Clock control register (Master mode) [3:0] (in _I2C_CCRH)

Definition at line 1594 of file STM8AF_STM8S.h.

5.1.2.629 _I2C_CCR0

```
#define _I2C_CCR0 ((uint8_t) (0x01 << 0))
```

I2C Clock control register (Master mode) [0] (in _I2C_CCRH)

Definition at line 1595 of file STM8AF_STM8S.h.

5.1.2.630 _I2C_CCR1

```
#define _I2C_CCR1 ((uint8_t) (0x01 << 1))
```

I2C Clock control register (Master mode) [1] (in _I2C_CCRH)

Definition at line 1596 of file STM8AF_STM8S.h.

5.1.2.631 _I2C_CCR2

```
#define _I2C_CCR2 ((uint8_t) (0x01 << 2))
```

I2C Clock control register (Master mode) [2] (in _I2C_CCRH)

Definition at line 1597 of file STM8AF_STM8S.h.

5.1.2.632 _I2C_CCR3

```
#define _I2C_CCR3 ((uint8_t) (0x01 << 3))
```

I2C Clock control register (Master mode) [3] (in _I2C_CCRH)

Definition at line 1598 of file STM8AF_STM8S.h.

5.1.2.633 _I2C_CCRH

```
#define _I2C_CCRH _SFR(uint8_t, I2C_AddressBase+0x0C)
```

I2C Clock control register high byte.

Definition at line 1498 of file STM8AF_STM8S.h.

5.1.2.634 _I2C_CCRH_RESET_VALUE

```
#define _I2C_CCRH_RESET_VALUE ((uint8_t) 0x00)
```

I2C Clock control register high byte reset value.

Definition at line 1514 of file STM8AF_STM8S.h.

5.1.2.635 _I2C_CCRL

```
#define _I2C_CCRL _SFR(uint8_t, I2C_AddressBase+0x0B)
```

I2C Clock control register low byte.

Definition at line 1497 of file STM8AF_STM8S.h.

5.1.2.636 _I2C_CCRL_RESET_VALUE

```
#define _I2C_CCRL_RESET_VALUE ((uint8_t) 0x00)
```

I2C Clock control register low byte reset value.

Definition at line 1513 of file STM8AF_STM8S.h.

5.1.2.637 _I2C_CR1

```
#define _I2C_CR1 _SFR(uint8_t, I2C_AddressBase+0x00)
```

I2C Control register 1.

Definition at line 1486 of file STM8AF_STM8S.h.

5.1.2.638 _I2C_CR1_RESET_VALUE

```
#define _I2C_CR1_RESET_VALUE ((uint8_t) 0x00)
```

I2C Control register 1 reset value.

Definition at line 1503 of file STM8AF_STM8S.h.

5.1.2.639 _I2C_CR2

```
#define _I2C_CR2 _SFR(uint8_t, I2C_AddressBase+0x01)
```

I2C Control register 2.

Definition at line 1487 of file STM8AF_STM8S.h.

5.1.2.640 _I2C_CR2_RESET_VALUE

```
#define _I2C_CR2_RESET_VALUE ((uint8_t) 0x00)
```

I2C Control register 2 reset value.

Definition at line 1504 of file STM8AF_STM8S.h.

5.1.2.641 _I2C_DR

```
#define _I2C_DR _SFR(uint8_t, I2C_AddressBase+0x06)
```

I2C data register.

Definition at line 1492 of file STM8AF_STM8S.h.

5.1.2.642 _I2C_DR_RESET_VALUE

```
#define _I2C_DR_RESET_VALUE ((uint8_t) 0x00)
```

I2C data register reset value.

Definition at line 1508 of file STM8AF_STM8S.h.

5.1.2.643 _I2C_DUTY

```
#define _I2C_DUTY ((uint8_t) (0x01 << 6))
```

I2C Fast mode duty cycle [0] (in _I2C_CCRH)

Definition at line 1600 of file STM8AF_STM8S.h.

5.1.2.644 _I2C_ENGC

```
#define _I2C_ENGC ((uint8_t) (0x01 << 6))
```

I2C General call enable [0] (in _I2C_CR1)

Definition at line 1520 of file STM8AF_STM8S.h.

5.1.2.645 _I2C_FREQ

```
#define _I2C_FREQ ((uint8_t) (0x3F << 0))
```

I2C Peripheral clock frequency [5:0] (in _I2C_FREQR)

Definition at line 1532 of file STM8AF_STM8S.h.

5.1.2.646 _I2C_FREQ0

```
#define _I2C_FREQ0 ((uint8_t) (0x01 << 0))
```

I2C Peripheral clock frequency [0] (in _I2C_FREQR)

Definition at line 1533 of file STM8AF_STM8S.h.

5.1.2.647 _I2C_FREQ1

```
#define _I2C_FREQ1 ((uint8_t) (0x01 << 1))
```

I2C Peripheral clock frequency [1] (in _I2C_FREQR)

Definition at line 1534 of file STM8AF_STM8S.h.

5.1.2.648 _I2C_FREQ2

```
#define _I2C_FREQ2 ((uint8_t) (0x01 << 2))
```

I2C Peripheral clock frequency [2] (in _I2C_FREQR)

Definition at line 1535 of file STM8AF_STM8S.h.

5.1.2.649 _I2C_FREQ3

```
#define _I2C_FREQ3 ((uint8_t) (0x01 << 3))
```

I2C Peripheral clock frequency [3] (in _I2C_FREQR)

Definition at line 1536 of file STM8AF_STM8S.h.

5.1.2.650 _I2C_FREQ4

```
#define _I2C_FREQ4 ((uint8_t) (0x01 << 4))
```

I2C Peripheral clock frequency [4] (in _I2C_FREQR)

Definition at line 1537 of file STM8AF_STM8S.h.

5.1.2.651 _I2C_FREQ5

```
#define _I2C_FREQ5 ((uint8_t) (0x01 << 5))
```

I2C Peripheral clock frequency [5] (in _I2C_FREQR)

Definition at line 1538 of file STM8AF_STM8S.h.

5.1.2.652 _I2C_FREQR

```
#define _I2C_FREQR _SFR(uint8_t, I2C_AddressBase+0x02)
```

I2C Frequency register.

Definition at line 1488 of file STM8AF_STM8S.h.

5.1.2.653 _I2C_FREQR_RESET_VALUE

```
#define _I2C_FREQR_RESET_VALUE ((uint8_t) 0x00)
```

I2C Frequency register reset value.

Definition at line 1505 of file STM8AF_STM8S.h.

5.1.2.654 _I2C_FS

```
#define _I2C_FS ((uint8_t) (0x01 << 7))
```

I2C Master mode selection [0] (in _I2C_CCRH)

Definition at line 1601 of file STM8AF_STM8S.h.

5.1.2.655 _I2C_GENCALL

```
#define _I2C_GENCALL ((uint8_t) (0x01 << 4))
```

I2C General call header (Slavemode) [0] (in _I2C_SR3)

Definition at line 1584 of file STM8AF_STM8S.h.

5.1.2.656 _I2C_ITBUFEN

```
#define _I2C_ITBUFEN ((uint8_t) (0x01 << 2))
```

I2C Buffer interrupt enable [0] (in _I2C_ITR)

Definition at line 1590 of file STM8AF_STM8S.h.

5.1.2.657 _I2C_ITERREN

```
#define _I2C_ITERREN ((uint8_t) (0x01 << 0))
```

I2C Error interrupt enable [0] (in _I2C_ITR)

Definition at line 1588 of file STM8AF_STM8S.h.

5.1.2.658 _I2C_ITEVTEN

```
#define _I2C_ITEVTEN ((uint8_t) (0x01 << 1))
```

I2C Event interrupt enable [0] (in _I2C_ITR)

Definition at line 1589 of file STM8AF_STM8S.h.

5.1.2.659 _I2C_ITR

```
#define _I2C_ITR _SFR(uint8_t, I2C_AddressBase+0x0A)
```

I2C Interrupt register.

Definition at line 1496 of file STM8AF_STM8S.h.

5.1.2.660 _I2C_ITR_RESET_VALUE

```
#define _I2C_ITR_RESET_VALUE ((uint8_t) 0x00)
```

I2C Interrupt register reset value.

Definition at line 1512 of file STM8AF_STM8S.h.

5.1.2.661 _I2C_MSL

```
#define _I2C_MSL ((uint8_t) (0x01 << 0))
```

I2C Master/Slave [0] (in _I2C_SR3)

Definition at line 1580 of file STM8AF_STM8S.h.

5.1.2.662 _I2C_NOSTRETCH

```
#define _I2C_NOSTRETCH ((uint8_t) (0x01 << 7))
```

I2C Clock stretching disable (Slave mode) [0] (in _I2C_CR1)

Definition at line 1521 of file STM8AF_STM8S.h.

5.1.2.663 _I2C_OARH

```
#define _I2C_OARH _SFR(uint8_t, I2C_AddressBase+0x04)
```

I2C own address register high byte.

Definition at line 1490 of file STM8AF_STM8S.h.

5.1.2.664 _I2C_OARH_RESET_VALUE

```
#define _I2C_OARH_RESET_VALUE ((uint8_t) 0x00)
```

I2C own address register high byte reset value.

Definition at line 1507 of file STM8AF_STM8S.h.

5.1.2.665 _I2C_OARL

```
#define _I2C_OARL _SFR(uint8_t, I2C_AddressBase+0x03)
```

I2C own address register low byte.

Definition at line 1489 of file STM8AF_STM8S.h.

5.1.2.666 _I2C_OARL_RESET_VALUE

```
#define _I2C_OARL_RESET_VALUE ((uint8_t) 0x00)
```

I2C own address register low byte reset value.

Definition at line 1506 of file STM8AF_STM8S.h.

5.1.2.667 _I2C_OVR

```
#define _I2C_OVR ((uint8_t) (0x01 << 3))
```

I2C Overrun/underrun [0] (in _I2C_SR2)

Definition at line 1574 of file STM8AF_STM8S.h.

5.1.2.668 _I2C_PE

```
#define _I2C_PE ((uint8_t) (0x01 << 0))
```

I2C Peripheral enable [0] (in _I2C_CR1)

Definition at line 1518 of file STM8AF_STM8S.h.

5.1.2.669 _I2C_POS

```
#define _I2C_POS ((uint8_t) (0x01 << 3))
```

I2C Acknowledge position (for data reception) [0] (in _I2C_CR2)

Definition at line 1527 of file STM8AF_STM8S.h.

5.1.2.670 _I2C_RXNE

```
#define _I2C_RXNE ((uint8_t) (0x01 << 6))
```

I2C Data register not empty (receivers) [0] (in _I2C_SR1)

Definition at line 1567 of file STM8AF_STM8S.h.

5.1.2.671 _I2C_SB

```
#define _I2C_SB ((uint8_t) (0x01 << 0))
```

I2C Start bit (Master mode) [0] (in _I2C_SR1)

Definition at line 1561 of file STM8AF_STM8S.h.

5.1.2.672 _I2C_SR1

```
#define _I2C_SR1 _SFR(uint8_t, I2C_AddressBase+0x07)
```

I2C Status register 1.

Definition at line 1493 of file STM8AF_STM8S.h.

5.1.2.673 _I2C_SR1_RESET_VALUE

```
#define _I2C_SR1_RESET_VALUE ((uint8_t) 0x00)
```

I2C Status register 1 reset value.

Definition at line 1509 of file STM8AF_STM8S.h.

5.1.2.674 _I2C_SR2

```
#define _I2C_SR2 _SFR(uint8_t, I2C_AddressBase+0x08)
```

I2C Status register 2.

Definition at line 1494 of file STM8AF_STM8S.h.

5.1.2.675 _I2C_SR2_RESET_VALUE

```
#define _I2C_SR2_RESET_VALUE ((uint8_t) 0x00)
```

I2C Status register 2 reset value.

Definition at line 1510 of file STM8AF_STM8S.h.

5.1.2.676 _I2C_SR3

```
#define _I2C_SR3 _SFR(uint8_t, I2C_AddressBase+0x09)
```

I2C Status register 3.

Definition at line 1495 of file STM8AF_STM8S.h.

5.1.2.677 _I2C_SR3_RESET_VALUE

```
#define _I2C_SR3_RESET_VALUE ((uint8_t) 0x00)
```

I2C Status register 3 reset value.

Definition at line 1511 of file STM8AF_STM8S.h.

5.1.2.678 _I2C_START

```
#define _I2C_START ((uint8_t) (0x01 << 0))
```

I2C Start generation [0] (in _I2C_CR2)

Definition at line 1524 of file STM8AF_STM8S.h.

5.1.2.679 _I2C_STOP

```
#define _I2C_STOP ((uint8_t) (0x01 << 1))
```

I2C Stop generation [0] (in _I2C_CR2)

Definition at line 1525 of file STM8AF_STM8S.h.

5.1.2.680 _I2C_STOPF

```
#define _I2C_STOPF ((uint8_t) (0x01 << 4))
```

I2C Stop detection (Slave mode) [0] (in _I2C_SR1)

Definition at line 1565 of file STM8AF_STM8S.h.

5.1.2.681 _I2C_SWRST

```
#define _I2C_SWRST ((uint8_t) (0x01 << 7))
```

I2C Software reset [0] (in _I2C_CR2)

Definition at line 1529 of file STM8AF_STM8S.h.

5.1.2.682 _I2C_TRA

```
#define _I2C_TRA ((uint8_t) (0x01 << 2))
```

I2C Transmitter/Receiver [0] (in _I2C_SR3)

Definition at line 1582 of file STM8AF_STM8S.h.

5.1.2.683 _I2C_TRISE

```
#define _I2C_TRISE ((uint8_t) (0x3F << 0))
```

I2C Maximum rise time (Master mode) [5:0] (in _I2C_TRISER)

Definition at line 1604 of file STM8AF_STM8S.h.

5.1.2.684 _I2C_TRISE0

```
#define _I2C_TRISE0 ((uint8_t) (0x01 << 0))
```

I2C Maximum rise time (Master mode) [0] (in _I2C_TRISER)

Definition at line 1605 of file STM8AF_STM8S.h.

5.1.2.685 _I2C_TRISE1

```
#define _I2C_TRISE1 ((uint8_t) (0x01 << 1))
```

I2C Maximum rise time (Master mode) [1] (in _I2C_TRISER)

Definition at line 1606 of file STM8AF_STM8S.h.

5.1.2.686 _I2C_TRISE2

```
#define _I2C_TRISE2 ((uint8_t) (0x01 << 2))
```

I2C Maximum rise time (Master mode) [2] (in _I2C_TRISER)

Definition at line 1607 of file STM8AF_STM8S.h.

5.1.2.687 _I2C_TRISE3

```
#define _I2C_TRISE3 ((uint8_t) (0x01 << 3))
```

I2C Maximum rise time (Master mode) [3] (in _I2C_TRISER)

Definition at line 1608 of file STM8AF_STM8S.h.

5.1.2.688 _I2C_TRISE4

```
#define _I2C_TRISE4 ((uint8_t) (0x01 << 4))
```

I2C Maximum rise time (Master mode) [4] (in _I2C_TRISER)

Definition at line 1609 of file STM8AF_STM8S.h.

5.1.2.689 _I2C_TRISE5

```
#define _I2C_TRISE5 ((uint8_t) (0x01 << 5))
```

I2C Maximum rise time (Master mode) [5] (in _I2C_TRISER)

Definition at line 1610 of file STM8AF_STM8S.h.

5.1.2.690 _I2C_TRISER

```
#define _I2C_TRISER _SFR(uint8_t, I2C_AddressBase+0x0D)
```

I2C rise time register.

Definition at line 1499 of file STM8AF_STM8S.h.

5.1.2.691 _I2C_TRISER_RESET_VALUE

```
#define _I2C_TRISER_RESET_VALUE ((uint8_t) 0x02)
```

I2C rise time register reset value.

Definition at line 1515 of file STM8AF_STM8S.h.

5.1.2.692 _I2C_TXE

```
#define _I2C_TXE ((uint8_t) (0x01 << 7))
```

I2C Data register empty (transmitters) [0] (in _I2C_SR1)

Definition at line 1568 of file STM8AF_STM8S.h.

5.1.2.693 _I2C_WUFH

```
#define _I2C_WUFH ((uint8_t) (0x01 << 5))
```

I2C Wakeup from Halt [0] (in _I2C_SR2)

Definition at line 1576 of file STM8AF_STM8S.h.

5.1.2.694 _ITC

```
#define _ITC _SFR(ITC_t, ITC_AddressBase)
```

ITC struct/bit access.

Definition at line 6401 of file STM8AF_STM8S.h.

5.1.2.695 _ITC_SPR1

```
#define _ITC_SPR1 _SFR(uint8_t, ITC_AddressBase+0x00)
```

Interrupt priority register 1/8.

Definition at line 6402 of file STM8AF_STM8S.h.

5.1.2.696 _ITC_SPR1_RESET_VALUE

```
#define _ITC_SPR1_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 1/8 reset value.

Definition at line 6412 of file STM8AF_STM8S.h.

5.1.2.697 _ITC_SPR2

```
#define _ITC_SPR2 _SFR(uint8_t, ITC_AddressBase+0x01)
```

Interrupt priority register 2/8.

Definition at line 6403 of file STM8AF_STM8S.h.

5.1.2.698 _ITC_SPR2_RESET_VALUE

```
#define _ITC_SPR2_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 2/8 reset value.

Definition at line 6413 of file STM8AF_STM8S.h.

5.1.2.699 _ITC_SPR3

```
#define _ITC_SPR3 _SFR(uint8_t, ITC_AddressBase+0x02)
```

Interrupt priority register 3/8.

Definition at line 6404 of file STM8AF_STM8S.h.

5.1.2.700 _ITC_SPR3_RESET_VALUE

```
#define _ITC_SPR3_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 3/8 reset value.

Definition at line 6414 of file STM8AF_STM8S.h.

5.1.2.701 _ITC_SPR4

```
#define _ITC_SPR4 _SFR(uint8_t, ITC_AddressBase+0x03)
```

Interrupt priority register 4/8.

Definition at line 6405 of file STM8AF_STM8S.h.

5.1.2.702 _ITC_SPR4_RESET_VALUE

```
#define _ITC_SPR4_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 4/8 reset value.

Definition at line 6415 of file STM8AF_STM8S.h.

5.1.2.703 _ITC_SPR5

```
#define _ITC_SPR5 _SFR(uint8_t, ITC_AddressBase+0x04)
```

Interrupt priority register 5/8.

Definition at line 6406 of file STM8AF_STM8S.h.

5.1.2.704 _ITC_SPR5_RESET_VALUE

```
#define _ITC_SPR5_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 5/8 reset value.

Definition at line 6416 of file STM8AF_STM8S.h.

5.1.2.705 _ITC_SPR6

```
#define _ITC_SPR6 _SFR(uint8_t, ITC_AddressBase+0x05)
```

Interrupt priority register 6/8.

Definition at line 6407 of file STM8AF_STM8S.h.

5.1.2.706 _ITC_SPR6_RESET_VALUE

```
#define _ITC_SPR6_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 6/8 reset value.

Definition at line 6417 of file STM8AF_STM8S.h.

5.1.2.707 _ITC_SPR7

```
#define _ITC_SPR7 _SFR(uint8_t, ITC_AddressBase+0x06)
```

Interrupt priority register 7/8.

Definition at line 6408 of file STM8AF_STM8S.h.

5.1.2.708 _ITC_SPR7_RESET_VALUE

```
#define _ITC_SPR7_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 7/8 reset value.

Definition at line 6418 of file STM8AF_STM8S.h.

5.1.2.709 _ITC_SPR8

```
#define _ITC_SPR8 _SFR(uint8_t, ITC_AddressBase+0x07)
```

Interrupt priority register 8/8.

Definition at line 6409 of file STM8AF_STM8S.h.

5.1.2.710 _ITC_SPR8_RESET_VALUE

```
#define _ITC_SPR8_RESET_VALUE ((uint8_t) 0x0F)
```

Interrupt priority register 8/8 reset value.

Definition at line 6419 of file STM8AF_STM8S.h.

5.1.2.711 _ITC_VECT10SPR

```
#define _ITC_VECT10SPR ((uint8_t) (0x03 << 4))
```

ITC interrupt priority vector 10 [1:0] (in _ITC_SPR3)

Definition at line 6454 of file STM8AF_STM8S.h.

5.1.2.712 _ITC_VECT10SPR0

```
#define _ITC_VECT10SPR0 ((uint8_t) (0x01 << 4))
```

ITC interrupt priority vector 10 [0] (in _ITC_SPR3)

Definition at line 6455 of file STM8AF_STM8S.h.

5.1.2.713 _ITC_VECT10SPR1

```
#define _ITC_VECT10SPR1 ((uint8_t) (0x01 << 5))
```

ITC interrupt priority vector 10 [1] (in _ITC_SPR3)

Definition at line 6456 of file STM8AF_STM8S.h.

5.1.2.714 _ITC_VECT11SPR

```
#define _ITC_VECT11SPR ((uint8_t) (0x03 << 6))
```

ITC interrupt priority vector 11 [1:0] (in _ITC_SPR3)

Definition at line 6457 of file STM8AF_STM8S.h.

5.1.2.715 _ITC_VECT11SPR0

```
#define _ITC_VECT11SPR0 ((uint8_t) (0x01 << 6))
```

ITC interrupt priority vector 11 [0] (in _ITC_SPR3)

Definition at line 6458 of file STM8AF_STM8S.h.

5.1.2.716 _ITC_VECT11SPR1

```
#define _ITC_VECT11SPR1 ((uint8_t) (0x01 << 7))
```

ITC interrupt priority vector 11 [1] (in _ITC_SPR3)

Definition at line 6459 of file STM8AF_STM8S.h.

5.1.2.717 _ITC_VECT12SPR

```
#define _ITC_VECT12SPR ((uint8_t) (0x03 << 0))
```

ITC interrupt priority vector 12 [1:0] (in _ITC_SPR4)

Definition at line 6462 of file STM8AF_STM8S.h.

5.1.2.718 _ITC_VECT12SPR0

```
#define _ITC_VECT12SPR0 ((uint8_t) (0x01 << 0))
```

ITC interrupt priority vector 12 [0] (in _ITC_SPR4)

Definition at line 6463 of file STM8AF_STM8S.h.

5.1.2.719 _ITC_VECT12SPR1

```
#define _ITC_VECT12SPR1 ((uint8_t) (0x01 << 1))
```

ITC interrupt priority vector 12 [1] (in _ITC_SPR4)

Definition at line 6464 of file STM8AF_STM8S.h.

5.1.2.720 _ITC_VECT13SPR

```
#define _ITC_VECT13SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 13 [1:0] (in _ITC_SPR4)

Definition at line 6465 of file STM8AF_STM8S.h.

5.1.2.721 _ITC_VECT13SPR0

```
#define _ITC_VECT13SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 13 [0] (in _ITC_SPR4)

Definition at line 6466 of file STM8AF_STM8S.h.

5.1.2.722 _ITC_VECT13SPR1

```
#define _ITC_VECT13SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 13 [1] (in _ITC_SPR4)

Definition at line 6467 of file STM8AF_STM8S.h.

5.1.2.723 _ITC_VECT14SPR

```
#define _ITC_VECT14SPR ((uint8_t) (0x03 << 4))
```

ITC interrupt priority vector 14 [1:0] (in _ITC_SPR4)

Definition at line 6468 of file STM8AF_STM8S.h.

5.1.2.724 _ITC_VECT14SPR0

```
#define _ITC_VECT14SPR0 ((uint8_t) (0x01 << 4))
```

ITC interrupt priority vector 14 [0] (in _ITC_SPR4)

Definition at line 6469 of file STM8AF_STM8S.h.

5.1.2.725 _ITC_VECT14SPR1

```
#define _ITC_VECT14SPR1 ((uint8_t) (0x01 << 5))
```

ITC interrupt priority vector 14 [1] (in _ITC_SPR4)

Definition at line 6470 of file STM8AF_STM8S.h.

5.1.2.726 _ITC_VECT15SPR

```
#define _ITC_VECT15SPR ((uint8_t) (0x03 << 6))
```

ITC interrupt priority vector 15 [1:0] (in _ITC_SPR4)

Definition at line 6471 of file STM8AF_STM8S.h.

5.1.2.727 _ITC_VECT15SPR0

```
#define _ITC_VECT15SPR0 ((uint8_t) (0x01 << 6))
```

ITC interrupt priority vector 15 [0] (in _ITC_SPR4)

Definition at line 6472 of file STM8AF_STM8S.h.

5.1.2.728 _ITC_VECT15SPR1

```
#define _ITC_VECT15SPR1 ((uint8_t) (0x01 << 7))
```

ITC interrupt priority vector 15 [1] (in _ITC_SPR4)

Definition at line 6473 of file STM8AF_STM8S.h.

5.1.2.729 _ITC_VECT16SPR

```
#define _ITC_VECT16SPR ((uint8_t) (0x03 << 0))
```

ITC interrupt priority vector 16 [1:0] (in _ITC_SPR5)

Definition at line 6476 of file STM8AF_STM8S.h.

5.1.2.730 _ITC_VECT16SPR0

```
#define _ITC_VECT16SPR0 ((uint8_t) (0x01 << 0))
```

ITC interrupt priority vector 16 [0] (in _ITC_SPR5)

Definition at line 6477 of file STM8AF_STM8S.h.

5.1.2.731 _ITC_VECT16SPR1

```
#define _ITC_VECT16SPR1 ((uint8_t) (0x01 << 1))
```

ITC interrupt priority vector 16 [1] (in _ITC_SPR5)

Definition at line 6478 of file STM8AF_STM8S.h.

5.1.2.732 _ITC_VECT17SPR

```
#define _ITC_VECT17SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 17 [1:0] (in _ITC_SPR5)

Definition at line 6479 of file STM8AF_STM8S.h.

5.1.2.733 _ITC_VECT17SPR0

```
#define _ITC_VECT17SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 17 [0] (in _ITC_SPR5)

Definition at line 6480 of file STM8AF_STM8S.h.

5.1.2.734 _ITC_VECT17SPR1

```
#define _ITC_VECT17SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 17 [1] (in _ITC_SPR5)

Definition at line 6481 of file STM8AF_STM8S.h.

5.1.2.735 _ITC_VECT18SPR

```
#define _ITC_VECT18SPR ((uint8_t) (0x03 << 4))
```

ITC interrupt priority vector 18 [1:0] (in _ITC_SPR5)

Definition at line 6482 of file STM8AF_STM8S.h.

5.1.2.736 _ITC_VECT18SPR0

```
#define _ITC_VECT18SPR0 ((uint8_t) (0x01 << 4))
```

ITC interrupt priority vector 18 [0] (in _ITC_SPR5)

Definition at line 6483 of file STM8AF_STM8S.h.

5.1.2.737 _ITC_VECT18SPR1

```
#define _ITC_VECT18SPR1 ((uint8_t) (0x01 << 5))
```

ITC interrupt priority vector 18 [1] (in _ITC_SPR5)

Definition at line 6484 of file STM8AF_STM8S.h.

5.1.2.738 _ITC_VECT19SPR

```
#define _ITC_VECT19SPR ((uint8_t) (0x03 << 6))
```

ITC interrupt priority vector 19 [1:0] (in _ITC_SPR5)

Definition at line 6485 of file STM8AF_STM8S.h.

5.1.2.739 _ITC_VECT19SPR0

```
#define _ITC_VECT19SPR0 ((uint8_t) (0x01 << 6))
```

ITC interrupt priority vector 19 [0] (in _ITC_SPR5)

Definition at line 6486 of file STM8AF_STM8S.h.

5.1.2.740 _ITC_VECT19SPR1

```
#define _ITC_VECT19SPR1 ((uint8_t) (0x01 << 7))
```

ITC interrupt priority vector 19 [1] (in _ITC_SPR5)

Definition at line 6487 of file STM8AF_STM8S.h.

5.1.2.741 _ITC_VECT1SPR

```
#define _ITC_VECT1SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 1 [1:0] (in _ITC_SPR1)

Definition at line 6423 of file STM8AF_STM8S.h.

5.1.2.742 _ITC_VECT1SPR0

```
#define _ITC_VECT1SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 1 [0] (in _ITC_SPR1)

Definition at line 6424 of file STM8AF_STM8S.h.

5.1.2.743 _ITC_VECT1SPR1

```
#define _ITC_VECT1SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 1 [1] (in _ITC_SPR1)

Definition at line 6425 of file STM8AF_STM8S.h.

5.1.2.744 _ITC_VECT20SPR

```
#define _ITC_VECT20SPR ((uint8_t) (0x03 << 0))
```

ITC interrupt priority vector 20 [1:0] (in _ITC_SPR6)

Definition at line 6490 of file STM8AF_STM8S.h.

5.1.2.745 _ITC_VECT20SPR0

```
#define _ITC_VECT20SPR0 ((uint8_t) (0x01 << 0))
```

ITC interrupt priority vector 20 [0] (in _ITC_SPR6)

Definition at line 6491 of file STM8AF_STM8S.h.

5.1.2.746 _ITC_VECT20SPR1

```
#define _ITC_VECT20SPR1 ((uint8_t) (0x01 << 1))
```

ITC interrupt priority vector 20 [1] (in _ITC_SPR6)

Definition at line 6492 of file STM8AF_STM8S.h.

5.1.2.747 _ITC_VECT21SPR

```
#define _ITC_VECT21SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 21 [1:0] (in _ITC_SPR6)

Definition at line 6493 of file STM8AF_STM8S.h.

5.1.2.748 _ITC_VECT21SPR0

```
#define _ITC_VECT21SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 21 [0] (in _ITC_SPR6)

Definition at line 6494 of file STM8AF_STM8S.h.

5.1.2.749 _ITC_VECT21SPR1

```
#define _ITC_VECT21SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 21 [1] (in _ITC_SPR6)

Definition at line 6495 of file STM8AF_STM8S.h.

5.1.2.750 _ITC_VECT22SPR

```
#define _ITC_VECT22SPR ((uint8_t) (0x03 << 4))
```

ITC interrupt priority vector 22 [1:0] (in _ITC_SPR6)

Definition at line 6496 of file STM8AF_STM8S.h.

5.1.2.751 _ITC_VECT22SPR0

```
#define _ITC_VECT22SPR0 ((uint8_t) (0x01 << 4))
```

ITC interrupt priority vector 22 [0] (in _ITC_SPR6)

Definition at line 6497 of file STM8AF_STM8S.h.

5.1.2.752 _ITC_VECT22SPR1

```
#define _ITC_VECT22SPR1 ((uint8_t) (0x01 << 5))
```

ITC interrupt priority vector 22 [1] (in _ITC_SPR6)

Definition at line 6498 of file STM8AF_STM8S.h.

5.1.2.753 _ITC_VECT23SPR

```
#define _ITC_VECT23SPR ((uint8_t) (0x03 << 6))
```

ITC interrupt priority vector 23 [1:0] (in _ITC_SPR6)

Definition at line 6499 of file STM8AF_STM8S.h.

5.1.2.754 _ITC_VECT23SPR0

```
#define _ITC_VECT23SPR0 ((uint8_t) (0x01 << 6))
```

ITC interrupt priority vector 23 [0] (in _ITC_SPR6)

Definition at line 6500 of file STM8AF_STM8S.h.

5.1.2.755 _ITC_VECT23SPR1

```
#define _ITC_VECT23SPR1 ((uint8_t) (0x01 << 7))
```

ITC interrupt priority vector 23 [1] (in _ITC_SPR6)

Definition at line 6501 of file STM8AF_STM8S.h.

5.1.2.756 _ITC_VECT24SPR

```
#define _ITC_VECT24SPR ((uint8_t) (0x03 << 0))
```

ITC interrupt priority vector 24 [1:0] (in _ITC_SPR7)

Definition at line 6504 of file STM8AF_STM8S.h.

5.1.2.757 _ITC_VECT24SPR0

```
#define _ITC_VECT24SPR0 ((uint8_t) (0x01 << 0))
```

ITC interrupt priority vector 24 [0] (in _ITC_SPR7)

Definition at line 6505 of file STM8AF_STM8S.h.

5.1.2.758 _ITC_VECT24SPR1

```
#define _ITC_VECT24SPR1 ((uint8_t) (0x01 << 1))
```

ITC interrupt priority vector 24 [1] (in _ITC_SPR7)

Definition at line 6506 of file STM8AF_STM8S.h.

5.1.2.759 _ITC_VECT25SPR

```
#define _ITC_VECT25SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 25 [1:0] (in _ITC_SPR7)

Definition at line 6507 of file STM8AF_STM8S.h.

5.1.2.760 _ITC_VECT25SPR0

```
#define _ITC_VECT25SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 25 [0] (in _ITC_SPR7)

Definition at line 6508 of file STM8AF_STM8S.h.

5.1.2.761 _ITC_VECT25SPR1

```
#define _ITC_VECT25SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 25 [1] (in _ITC_SPR7)

Definition at line 6509 of file STM8AF_STM8S.h.

5.1.2.762 _ITC_VECT26SPR

```
#define _ITC_VECT26SPR ((uint8_t) (0x03 << 4))
```

ITC interrupt priority vector 26 [1:0] (in _ITC_SPR7)

Definition at line 6510 of file STM8AF_STM8S.h.

5.1.2.763 _ITC_VECT26SPR0

```
#define _ITC_VECT26SPR0 ((uint8_t) (0x01 << 4))
```

ITC interrupt priority vector 26 [0] (in _ITC_SPR7)

Definition at line 6511 of file STM8AF_STM8S.h.

5.1.2.764 _ITC_VECT26SPR1

```
#define _ITC_VECT26SPR1 ((uint8_t) (0x01 << 5))
```

ITC interrupt priority vector 26 [1] (in _ITC_SPR7)

Definition at line 6512 of file STM8AF_STM8S.h.

5.1.2.765 _ITC_VECT27SPR

```
#define _ITC_VECT27SPR ((uint8_t) (0x03 << 6))
```

ITC interrupt priority vector 27 [1:0] (in _ITC_SPR7)

Definition at line 6513 of file STM8AF_STM8S.h.

5.1.2.766 _ITC_VECT27SPR0

```
#define _ITC_VECT27SPR0 ((uint8_t) (0x01 << 6))
```

ITC interrupt priority vector 27 [0] (in _ITC_SPR7)

Definition at line 6514 of file STM8AF_STM8S.h.

5.1.2.767 _ITC_VECT27SPR1

```
#define _ITC_VECT27SPR1 ((uint8_t) (0x01 << 7))
```

ITC interrupt priority vector 27 [1] (in _ITC_SPR7)

Definition at line 6515 of file STM8AF_STM8S.h.

5.1.2.768 _ITC_VECT28SPR

```
#define _ITC_VECT28SPR ((uint8_t) (0x03 << 0))
```

ITC interrupt priority vector 28 [1:0] (in _ITC_SPR8)

Definition at line 6518 of file STM8AF_STM8S.h.

5.1.2.769 _ITC_VECT28SPR0

```
#define _ITC_VECT28SPR0 ((uint8_t) (0x01 << 0))
```

ITC interrupt priority vector 28 [0] (in _ITC_SPR8)

Definition at line 6519 of file STM8AF_STM8S.h.

5.1.2.770 _ITC_VECT28SPR1

```
#define _ITC_VECT28SPR1 ((uint8_t) (0x01 << 1))
```

ITC interrupt priority vector 28 [1] (in _ITC_SPR8)

Definition at line 6520 of file STM8AF_STM8S.h.

5.1.2.771 _ITC_VECT29SPR

```
#define _ITC_VECT29SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 29 [1:0] (in _ITC_SPR8)

Definition at line 6521 of file STM8AF_STM8S.h.

5.1.2.772 _ITC_VECT29SPR0

```
#define _ITC_VECT29SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 29 [0] (in _ITC_SPR8)

Definition at line 6522 of file STM8AF_STM8S.h.

5.1.2.773 _ITC_VECT29SPR1

```
#define _ITC_VECT29SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 29 [1] (in _ITC_SPR8)

Definition at line 6523 of file STM8AF_STM8S.h.

5.1.2.774 _ITC_VECT2SPR

```
#define _ITC_VECT2SPR ((uint8_t) (0x03 << 4))
```

ITC interrupt priority vector 2 [1:0] (in _ITC_SPR1)

Definition at line 6426 of file STM8AF_STM8S.h.

5.1.2.775 _ITC_VECT2SPR0

```
#define _ITC_VECT2SPR0 ((uint8_t) (0x01 << 4))
```

ITC interrupt priority vector 2 [0] (in _ITC_SPR1)

Definition at line 6427 of file STM8AF_STM8S.h.

5.1.2.776 _ITC_VECT2SPR1

```
#define _ITC_VECT2SPR1 ((uint8_t) (0x01 << 5))
```

ITC interrupt priority vector 2 [1] (in _ITC_SPR1)

Definition at line 6428 of file STM8AF_STM8S.h.

5.1.2.777 _ITC_VECT3SPR

```
#define _ITC_VECT3SPR ((uint8_t) (0x03 << 6))
```

ITC interrupt priority vector 3 [1:0] (in _ITC_SPR1)

Definition at line 6429 of file STM8AF_STM8S.h.

5.1.2.778 _ITC_VECT3SPR0

```
#define _ITC_VECT3SPR0 ((uint8_t) (0x01 << 6))
```

ITC interrupt priority vector 3 [0] (in _ITC_SPR1)

Definition at line 6430 of file STM8AF_STM8S.h.

5.1.2.779 _ITC_VECT3SPR1

```
#define _ITC_VECT3SPR1 ((uint8_t) (0x01 << 7))
```

ITC interrupt priority vector 3 [1] (in _ITC_SPR1)

Definition at line 6431 of file STM8AF_STM8S.h.

5.1.2.780 _ITC_VECT4SPR

```
#define _ITC_VECT4SPR ((uint8_t) (0x03 << 0))
```

ITC interrupt priority vector 4 [1:0] (in _ITC_SPR2)

Definition at line 6434 of file STM8AF_STM8S.h.

5.1.2.781 _ITC_VECT4SPR0

```
#define _ITC_VECT4SPR0 ((uint8_t) (0x01 << 0))
```

ITC interrupt priority vector 4 [0] (in _ITC_SPR2)

Definition at line 6435 of file STM8AF_STM8S.h.

5.1.2.782 _ITC_VECT4SPR1

```
#define _ITC_VECT4SPR1 ((uint8_t) (0x01 << 1))
```

ITC interrupt priority vector 4 [1] (in _ITC_SPR2)

Definition at line 6436 of file STM8AF_STM8S.h.

5.1.2.783 _ITC_VECT5SPR

```
#define _ITC_VECT5SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 5 [1:0] (in _ITC_SPR2)

Definition at line 6437 of file STM8AF_STM8S.h.

5.1.2.784 _ITC_VECT5SPR0

```
#define _ITC_VECT5SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 5 [0] (in _ITC_SPR2)

Definition at line 6438 of file STM8AF_STM8S.h.

5.1.2.785 _ITC_VECT5SPR1

```
#define _ITC_VECT5SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 5 [1] (in _ITC_SPR2)

Definition at line 6439 of file STM8AF_STM8S.h.

5.1.2.786 _ITC_VECT6SPR

```
#define _ITC_VECT6SPR ((uint8_t) (0x03 << 4))
```

ITC interrupt priority vector 6 [1:0] (in _ITC_SPR2)

Definition at line 6440 of file STM8AF_STM8S.h.

5.1.2.787 _ITC_VECT6SPR0

```
#define _ITC_VECT6SPR0 ((uint8_t) (0x01 << 4))
```

ITC interrupt priority vector 6 [0] (in _ITC_SPR2)

Definition at line 6441 of file STM8AF_STM8S.h.

5.1.2.788 _ITC_VECT6SPR1

```
#define _ITC_VECT6SPR1 ((uint8_t) (0x01 << 5))
```

ITC interrupt priority vector 6 [1] (in _ITC_SPR2)

Definition at line 6442 of file STM8AF_STM8S.h.

5.1.2.789 _ITC_VECT7SPR

```
#define _ITC_VECT7SPR ((uint8_t) (0x03 << 6))
```

ITC interrupt priority vector 7 [1:0] (in _ITC_SPR2)

Definition at line 6443 of file STM8AF_STM8S.h.

5.1.2.790 _ITC_VECT7SPR0

```
#define _ITC_VECT7SPR0 ((uint8_t) (0x01 << 6))
```

ITC interrupt priority vector 7 [0] (in _ITC_SPR2)

Definition at line 6444 of file STM8AF_STM8S.h.

5.1.2.791 _ITC_VECT7SPR1

```
#define _ITC_VECT7SPR1 ((uint8_t) (0x01 << 7))
```

ITC interrupt priority vector 7 [1] (in _ITC_SPR2)

Definition at line 6445 of file STM8AF_STM8S.h.

5.1.2.792 _ITC_VECT8SPR

```
#define _ITC_VECT8SPR ((uint8_t) (0x03 << 0))
```

ITC interrupt priority vector 8 [1:0] (in _ITC_SPR3)

Definition at line 6448 of file STM8AF_STM8S.h.

5.1.2.793 _ITC_VECT8SPR0

```
#define _ITC_VECT8SPR0 ((uint8_t) (0x01 << 0))
```

ITC interrupt priority vector 8 [0] (in _ITC_SPR3)

Definition at line 6449 of file STM8AF_STM8S.h.

5.1.2.794 _ITC_VECT8SPR1

```
#define _ITC_VECT8SPR1 ((uint8_t) (0x01 << 1))
```

ITC interrupt priority vector 8 [1] (in _ITC_SPR3)

Definition at line 6450 of file STM8AF_STM8S.h.

5.1.2.795 _ITC_VECT9SPR

```
#define _ITC_VECT9SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 9 [1:0] (in _ITC_SPR3)

Definition at line 6451 of file STM8AF_STM8S.h.

5.1.2.796 _ITC_VECT9SPR0

```
#define _ITC_VECT9SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 9 [0] (in _ITC_SPR3)

Definition at line 6452 of file STM8AF_STM8S.h.

5.1.2.797 _ITC_VECT9SPR1

```
#define _ITC_VECT9SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 9 [1] (in _ITC_SPR3)

Definition at line 6453 of file STM8AF_STM8S.h.

5.1.2.798 _IWDG

```
#define _IWDG _SFR(IWDG_t, IWDG_AddressBase)
```

Independent Timeout Watchdog struct/bit access.

Definition at line 1069 of file STM8AF_STM8S.h.

5.1.2.799 _IWDG_KEY_ACCESS

```
#define _IWDG_KEY_ACCESS ((uint8_t) 0x55)
```

Independent Timeout Watchdog unlock write to IWDG_PR and IWDG_RLR (in _IWDG_KR)

Definition at line 1081 of file STM8AF_STM8S.h.

5.1.2.800 _IWDG_KEY_ENABLE

```
#define _IWDG_KEY_ENABLE ((uint8_t) 0xCC)
```

Independent Timeout Watchdog enable (in _IWDG_KR)

Definition at line 1079 of file STM8AF_STM8S.h.

5.1.2.801 _IWDG_KEY_REFRESH

```
#define _IWDG_KEY_REFRESH ((uint8_t) 0xAA)
```

Independent Timeout Watchdog refresh (in _IWDG_KR)

Definition at line 1080 of file STM8AF_STM8S.h.

5.1.2.802 _IWDG_KR

```
#define _IWDG_KR _SFR(uint8_t, IWDG_AddressBase+0x00)
```

Independent Timeout Watchdog Key register (IWDG_KR)

Definition at line 1070 of file STM8AF_STM8S.h.

5.1.2.803 _IWDG_PR

```
#define _IWDG_PR _SFR(uint8_t, IWDG_AddressBase+0x01)
```

Independent Timeout Watchdog Prescaler register (IWDG_PR)

Definition at line 1071 of file STM8AF_STM8S.h.

5.1.2.804 _IWDG_PR_RESET_VALUE

```
#define _IWDG_PR_RESET_VALUE ((uint8_t) 0x00)
```

Independent Timeout Watchdog Prescaler register reset value.

Definition at line 1075 of file STM8AF_STM8S.h.

5.1.2.805 _IWDG_PRE

```
#define _IWDG_PRE ((uint8_t) (0x07 << 0))
```

Independent Timeout Watchdog Prescaler divider [2:0] (in _IWDG_PR)

Definition at line 1084 of file STM8AF_STM8S.h.

5.1.2.806 _IWDG_PRE0

```
#define _IWDG_PRE0 ((uint8_t) (0x01 << 0))
```

Independent Timeout Watchdog Prescaler divider [0] (in _IWDG_PR)

Definition at line 1085 of file STM8AF_STM8S.h.

5.1.2.807 _IWDG_PRE1

```
#define _IWDG_PRE1 ((uint8_t) (0x01 << 1))
```

Independent Timeout Watchdog Prescaler divider [1] (in _IWDG_PR)

Definition at line 1086 of file STM8AF_STM8S.h.

5.1.2.808 _IWDG_PRE2

```
#define _IWDG_PRE2 ((uint8_t) (0x01 << 2))
```

Independent Timeout Watchdog Prescaler divider [2] (in _IWDG_PR)

Definition at line 1087 of file STM8AF_STM8S.h.

5.1.2.809 _IWDG_RLR

```
#define _IWDG_RLR _SFR(uint8_t, IWDG_AddressBase+0x02)
```

Independent Timeout Watchdog Reload register (IWDG_RLR)

Definition at line 1072 of file STM8AF_STM8S.h.

5.1.2.810 _IWDG_RLR_RESET_VALUE

```
#define _IWDG_RLR_RESET_VALUE ((uint8_t) 0xFF)
```

Independent Timeout Watchdog Reload register reset value.

Definition at line 1076 of file STM8AF_STM8S.h.

5.1.2.811 _RST

```
#define _RST _SFR(RST_t, RST_AddressBase)
```

Reset module struct/bit access.

Definition at line 725 of file STM8AF_STM8S.h.

5.1.2.812 _RST_EMCF

```
#define _RST_EMCF ((uint8_t) (0x01 << 4))
```

EMC reset flag [0] (in _RST_SR)

Definition at line 733 of file STM8AF_STM8S.h.

5.1.2.813 _RST_ILLOPF

```
#define _RST_ILLOPF ((uint8_t) (0x01 << 2))
```

Illegal opcode reset flag [0] (in _RST_SR)

Definition at line 731 of file STM8AF_STM8S.h.

5.1.2.814 _RST_IWDGF

```
#define _RST_IWDGF ((uint8_t) (0x01 << 1))
```

Independent Watchdog reset flag [0] (in _RST_SR)

Definition at line 730 of file STM8AF_STM8S.h.

5.1.2.815 _RST_SR

```
#define _RST_SR _SFR(uint8_t, RST_AddressBase+0x00)
```

Reset module status register (RST_SR)

Definition at line 726 of file STM8AF_STM8S.h.

5.1.2.816 _RST_SWIMF

```
#define _RST_SWIMF ((uint8_t) (0x01 << 3))
```

SWIM reset flag [0] (in _RST_SR)

Definition at line 732 of file STM8AF_STM8S.h.

5.1.2.817 _RST_WWDGF

```
#define _RST_WWDGF ((uint8_t) (0x01 << 0))
```

Window Watchdog reset flag [0] (in _RST_SR)

Definition at line 729 of file STM8AF_STM8S.h.

5.1.2.818 _SFR

```
#define _SFR(
              type,
              addr ) (*((volatile type*) (addr)))
```

peripheral register

Definition at line 187 of file STM8AF_STM8S.h.

5.1.2.819 _SPI

```
#define _SPI _SFR(SPI_t, SPI_AddressBase)
```

register for SPI control

SPI struct/bit access

Definition at line 1288 of file STM8AF_STM8S.h.

5.1.2.820 _SPI_BDM

```
#define _SPI_BDM ((uint8_t) (0x01 << 7))
```

SPI Bidirectional data mode enable [0] (in _SPI_CR2)

Definition at line 1327 of file STM8AF_STM8S.h.

5.1.2.821 _SPI_BDOE

```
#define _SPI_BDOE ((uint8_t) (0x01 << 6))
```

SPI Input/Output enable in bidirectional mode [0] (in _SPI_CR2)

Definition at line 1326 of file STM8AF_STM8S.h.

5.1.2.822 _SPI_BR

```
#define _SPI_BR ((uint8_t) (0x07 << 3))
```

SPI Baudrate control [2:0] (in _SPI_CR1)

Definition at line 1312 of file STM8AF_STM8S.h.

5.1.2.823 _SPI_BR0

```
#define _SPI_BR0 ((uint8_t) (0x01 << 3))
```

SPI Baudrate control [0] (in _SPI_CR1)

Definition at line 1313 of file STM8AF_STM8S.h.

5.1.2.824 _SPI_BR1

```
#define _SPI_BR1 ((uint8_t) (0x01 << 4))
```

SPI Baudrate control [1] (in _SPI_CR1)

Definition at line 1314 of file STM8AF_STM8S.h.

5.1.2.825 _SPI_BR2

```
#define _SPI_BR2 ((uint8_t) (0x01 << 5))
```

SPI Baudrate control [2] (in _SPI_CR1)

Definition at line 1315 of file STM8AF_STM8S.h.

5.1.2.826 _SPI_BSY

```
#define _SPI_BSY ((uint8_t) (0x01 << 7))
```

SPI Busy flag [0] (in _SPI_SR)

Definition at line 1344 of file STM8AF_STM8S.h.

5.1.2.827 _SPI_CPHA

```
#define _SPI_CPHA ((uint8_t) (0x01 << 0))
```

SPI Clock phase [0] (in _SPI_CR1)

Definition at line 1309 of file STM8AF_STM8S.h.

5.1.2.828 _SPI_CPOL

```
#define _SPI_CPOL ((uint8_t) (0x01 << 1))
```

SPI Clock polarity [0] (in _SPI_CR1)

Definition at line 1310 of file STM8AF_STM8S.h.

5.1.2.829 _SPI_CR1

```
#define _SPI_CR1 _SFR(uint8_t, SPI_AddressBase+0x00)
```

SPI control register 1.

Definition at line 1289 of file STM8AF_STM8S.h.

5.1.2.830 _SPI_CR1_RESET_VALUE

```
#define _SPI_CR1_RESET_VALUE ((uint8_t) 0x00)
```

SPI Control Register 1 reset value.

Definition at line 1299 of file STM8AF_STM8S.h.

5.1.2.831 _SPI_CR2

```
#define _SPI_CR2 _SFR(uint8_t, SPI_AddressBase+0x01)
```

SPI control register 2.

Definition at line 1290 of file STM8AF_STM8S.h.

5.1.2.832 _SPI_CR2_RESET_VALUE

```
#define _SPI_CR2_RESET_VALUE ((uint8_t) 0x00)
```

SPI Control Register 2 reset value.

Definition at line 1300 of file STM8AF_STM8S.h.

5.1.2.833 _SPI_CRCEN

```
#define _SPI_CRCEN ((uint8_t) (0x01 << 5))
```

SPI Hardware CRC calculation enable [0] (in _SPI_CR2)

Definition at line 1325 of file STM8AF_STM8S.h.

5.1.2.834 _SPI_CRCERR

```
#define _SPI_CRCERR ((uint8_t) (0x01 << 4))
```

SPI CRC error flag [0] (in _SPI_SR)

Definition at line 1341 of file STM8AF_STM8S.h.

5.1.2.835 _SPI_CRCNEXT

```
#define _SPI_CRCNEXT ((uint8_t) (0x01 << 4))
```

SPI Transmit CRC next [0] (in _SPI_CR2)

Definition at line 1324 of file STM8AF_STM8S.h.

5.1.2.836 _SPI_CRCPR

```
#define _SPI_CRCPR _SFR(uint8_t, SPI_AddressBase+0x05)
```

SPI CRC polynomial register.

Definition at line 1294 of file STM8AF_STM8S.h.

5.1.2.837 _SPI_CRCPR_RESET_VALUE

```
#define _SPI_CRCPR_RESET_VALUE ((uint8_t) 0x07)
```

SPI Polynomial Register reset value.

Definition at line 1304 of file STM8AF_STM8S.h.

5.1.2.838 _SPI_DR

```
#define _SPI_DR _SFR(uint8_t, SPI_AddressBase+0x04)
```

SPI data register.

Definition at line 1293 of file STM8AF_STM8S.h.

5.1.2.839 _SPI_DR_RESET_VALUE

```
#define _SPI_DR_RESET_VALUE ((uint8_t) 0x00)
```

SPI Data Register reset value.

Definition at line 1303 of file STM8AF_STM8S.h.

5.1.2.840 _SPI_ERRIE

```
#define _SPI_ERRIE ((uint8_t) (0x01 << 5))
```

SPI Error interrupt enable [0] (in _SPI_ICR)

Definition at line 1332 of file STM8AF_STM8S.h.

5.1.2.841 _SPI_ICR

```
#define _SPI_ICR _SFR(uint8_t, SPI_AddressBase+0x02)
```

SPI interrupt control register.

Definition at line 1291 of file STM8AF_STM8S.h.

5.1.2.842 _SPI_ICR_RESET_VALUE

```
#define _SPI_ICR_RESET_VALUE ((uint8_t) 0x00)
```

SPI Interrupt Control Register reset value.

Definition at line 1301 of file STM8AF_STM8S.h.

5.1.2.843 _SPI_LSBFIRST

```
#define _SPI_LSBFIRST ((uint8_t) (0x01 << 7))
```

SPI Frame format [0] (in _SPI_CR1)

Definition at line 1317 of file STM8AF_STM8S.h.

5.1.2.844 _SPI_MODF

```
#define _SPI_MODF ((uint8_t) (0x01 << 5))
```

SPI Mode fault [0] (in _SPI_SR)

Definition at line 1342 of file STM8AF_STM8S.h.

5.1.2.845 _SPI_MSTR

```
#define _SPI_MSTR ((uint8_t) (0x01 << 2))
```

SPI Master/slave selection [0] (in _SPI_CR1)

Definition at line 1311 of file STM8AF_STM8S.h.

5.1.2.846 _SPI_OVR

```
#define _SPI_OVR ((uint8_t) (0x01 << 6))
```

SPI Overrun flag [0] (in _SPI_SR)

Definition at line 1343 of file STM8AF_STM8S.h.

5.1.2.847 _SPI_RXCRCR

```
#define _SPI_RXCRCR _SFR(uint8_t, SPI_AddressBase+0x06)
```

SPI Rx CRC register.

Definition at line 1295 of file STM8AF_STM8S.h.

5.1.2.848 _SPI_RXCRCR_RESET_VALUE

```
#define _SPI_RXCRCR_RESET_VALUE ((uint8_t) 0x00)
```

SPI RX CRC Register reset value.

Definition at line 1305 of file STM8AF_STM8S.h.

5.1.2.849 _SPI_RXIE

```
#define _SPI_RXIE ((uint8_t) (0x01 << 6))
```

SPI Rx buffer not empty interrupt enable [0] (in _SPI_ICR)

Definition at line 1333 of file STM8AF_STM8S.h.

5.1.2.850 _SPI_RXNE

```
#define _SPI_RXNE ((uint8_t) (0x01 << 0))
```

SPI Receive buffer not empty [0] (in _SPI_SR)

Definition at line 1337 of file STM8AF_STM8S.h.

5.1.2.851 _SPI_RXONLY

```
#define _SPI_RXONLY ((uint8_t) (0x01 << 2))
```

SPI Receive only [0] (in _SPI_CR2)

Definition at line 1322 of file STM8AF_STM8S.h.

5.1.2.852 _SPI_SPE

```
#define _SPI_SPE ((uint8_t) (0x01 << 6))
```

SPI enable [0] (in _SPI_CR1)

Definition at line 1316 of file STM8AF_STM8S.h.

5.1.2.853 _SPI_SR

```
#define _SPI_SR _SFR(uint8_t, SPI_AddressBase+0x03)
```

SPI status register.

Definition at line 1292 of file STM8AF_STM8S.h.

5.1.2.854 _SPI_SR_RESET_VALUE

```
#define _SPI_SR_RESET_VALUE ((uint8_t) 0x02)
```

SPI Status Register reset value.

Definition at line 1302 of file STM8AF_STM8S.h.

5.1.2.855 _SPI_SSI

```
#define _SPI_SSI ((uint8_t) (0x01 << 0))
```

SPI Internal slave select [0] (in _SPI_CR2)

Definition at line 1320 of file STM8AF_STM8S.h.

5.1.2.856 _SPI_SSM

```
#define _SPI_SSM ((uint8_t) (0x01 << 1))
```

SPI Software slave management [0] (in _SPI_CR2)

Definition at line 1321 of file STM8AF_STM8S.h.

5.1.2.857 _SPI_TXCRCR

```
#define _SPI_TXCRCR _SFR(uint8_t, SPI_AddressBase+0x07)
```

SPI Tx CRC register.

Definition at line 1296 of file STM8AF_STM8S.h.

5.1.2.858 _SPI_TXCRCR_RESET_VALUE

```
#define _SPI_TXCRCR_RESET_VALUE ((uint8_t) 0x00)
```

SPI TX CRC Register reset value.

Definition at line 1306 of file STM8AF_STM8S.h.

5.1.2.859 _SPI_TXE

```
#define _SPI_TXE ((uint8_t) (0x01 << 1))
```

SPI Transmit buffer empty [0] (in _SPI_SR)

Definition at line 1338 of file STM8AF_STM8S.h.

5.1.2.860 _SPI_TXIE

```
#define _SPI_TXIE ((uint8_t) (0x01 << 7))
```

SPI Tx buffer empty interrupt enable [0] (in _SPI_ICR)

Definition at line 1334 of file STM8AF_STM8S.h.

5.1.2.861 _SPI_WKIE

```
#define _SPI_WKIE ((uint8_t) (0x01 << 4))
```

SPI Wakeup interrupt enable [0] (in _SPI_ICR)

Definition at line 1331 of file STM8AF_STM8S.h.

5.1.2.862 _SPI_WKUP

```
#define _SPI_WKUP ((uint8_t) (0x01 << 3))
```

SPI Wakeup flag [0] (in _SPI_SR)

Definition at line 1340 of file STM8AF_STM8S.h.

5.1.2.863 _TIM1

```
#define _TIM1 _SFR(TIM1_t, TIM1_AddressBase)
```

TIM1 struct/bit access.

Definition at line 2803 of file STM8AF_STM8S.h.

5.1.2.864 _TIM1_AOE

```
#define _TIM1_AOE ((uint8_t) (0x01 << 6))
```

TIM1 Automatic output enable [0] (in _TIM1_BKR)

Definition at line 3074 of file STM8AF_STM8S.h.

5.1.2.865 _TIM1_ARPE

```
#define _TIM1_ARPE ((uint8_t) (0x01 << 7))
```

TIM1 Auto-reload preload enable [0] (in _TIM1_CR1)

Definition at line 2880 of file STM8AF_STM8S.h.

5.1.2.866 _TIM1_ARRH

```
#define _TIM1_ARRH _SFR(uint8_t, TIM1_AddressBase+0x12)
```

TIM1 auto-reload register high byte.

Definition at line 2822 of file STM8AF_STM8S.h.

5.1.2.867 _TIM1_ARRH_RESET_VALUE

```
#define _TIM1_ARRH_RESET_VALUE ((uint8_t) 0xFF)
```

TIM1 auto-reload register high byte reset value.

Definition at line 2856 of file STM8AF_STM8S.h.

5.1.2.868 _TIM1_ARRL

```
#define _TIM1_ARRL _SFR(uint8_t, TIM1_AddressBase+0x13)
```

TIM1 auto-reload register low byte.

Definition at line 2823 of file STM8AF_STM8S.h.

5.1.2.869 _TIM1_ARRL_RESET_VALUE

```
#define _TIM1_ARRL_RESET_VALUE ((uint8_t) 0xFF)
```

TIM1 auto-reload register low byte reset value.

Definition at line 2857 of file STM8AF_STM8S.h.

5.1.2.870 _TIM1_BG

```
#define _TIM1_BG ((uint8_t) (0x01 << 7))
```

TIM1 Break generation [0] (in _TIM1_EGR)

Definition at line 2953 of file STM8AF_STM8S.h.

5.1.2.871 _TIM1_BIE

```
#define _TIM1_BIE ((uint8_t) (0x01 << 7))
```

TIM1 Break interrupt enable [0] (in _TIM1_IER)

Definition at line 2925 of file STM8AF_STM8S.h.

5.1.2.872 _TIM1_BIF

```
#define _TIM1_BIF ((uint8_t) (0x01 << 7))
```

TIM1 Break interrupt flag [0] (in _TIM1_SR1)

Definition at line 2935 of file STM8AF_STM8S.h.

5.1.2.873 _TIM1_BKE

```
#define _TIM1_BKE ((uint8_t) (0x01 << 4))
```

TIM1 Break enable [0] (in _TIM1_BKR)

Definition at line 3072 of file STM8AF_STM8S.h.

5.1.2.874 _TIM1_BKP

```
#define _TIM1_BKP ((uint8_t) (0x01 << 5))
```

TIM1 Break polarity [0] (in _TIM1_BKR)

Definition at line 3073 of file STM8AF_STM8S.h.

5.1.2.875 _TIM1_BKR

```
#define _TIM1_BKR _SFR(uint8_t, TIM1_AddressBase+0x1D)
```

TIM1 Break register.

Definition at line 2833 of file STM8AF_STM8S.h.

5.1.2.876 _TIM1_BKR_RESET_VALUE

```
#define _TIM1_BKR_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 Break register reset value.

Definition at line 2867 of file STM8AF_STM8S.h.

5.1.2.877 _TIM1_CC1E

```
#define _TIM1_CC1E ((uint8_t) (0x01 << 0))
```

TIM1 Capture/compare 1 output enable [0] (in _TIM1_CCER1)

Definition at line 3048 of file STM8AF_STM8S.h.

5.1.2.878 _TIM1_CC1G

```
#define _TIM1_CC1G ((uint8_t) (0x01 << 1))
```

TIM1 Capture/compare 1 generation [0] (in _TIM1_EGR)

Definition at line 2947 of file STM8AF_STM8S.h.

5.1.2.879 _TIM1_CC1IE

```
#define _TIM1_CC1IE ((uint8_t) (0x01 << 1))
```

TIM1 Capture/compare 1 interrupt enable [0] (in _TIM1_IER)

Definition at line 2919 of file STM8AF_STM8S.h.

5.1.2.880 _TIM1_CC1IF

```
#define _TIM1_CC1IF ((uint8_t) (0x01 << 1))
```

TIM1 Capture/compare 1 interrupt flag [0] (in _TIM1_SR1)

Definition at line 2929 of file STM8AF_STM8S.h.

5.1.2.881 _TIM1_CC1NE

```
#define _TIM1_CC1NE ((uint8_t) (0x01 << 2))
```

TIM1 Capture/compare 1 complementary output enable [0] (in _TIM1_CCER1)

Definition at line 3050 of file STM8AF_STM8S.h.

5.1.2.882 _TIM1_CC1NP

```
#define _TIM1_CC1NP ((uint8_t) (0x01 << 3))
```

TIM1 Capture/compare 1 complementary output polarity [0] (in _TIM1_CCER1)

Definition at line 3051 of file STM8AF_STM8S.h.

5.1.2.883 _TIM1_CC1OF

```
#define _TIM1_CC1OF ((uint8_t) (0x01 << 1))
```

TIM1 Capture/compare 1 overcapture flag [0] (in _TIM1_SR2)

Definition at line 2939 of file STM8AF_STM8S.h.

5.1.2.884 _TIM1_CC1P

```
#define _TIM1_CC1P ((uint8_t) (0x01 << 1))
```

TIM1 Capture/compare 1 output polarity [0] (in _TIM1_CCER1)

Definition at line 3049 of file STM8AF_STM8S.h.

5.1.2.885 _TIM1_CC1S

```
#define _TIM1_CC1S ((uint8_t) (0x03 << 0))
```

TIM1 Compare 1 selection [1:0] (in _TIM1_CCMR1)

Definition at line 2956 of file STM8AF_STM8S.h.

5.1.2.886 _TIM1_CC1S0

```
#define _TIM1_CC1S0 ((uint8_t) (0x01 << 0))
```

TIM1 Compare 1 selection [0] (in _TIM1_CCMR1)

Definition at line 2957 of file STM8AF_STM8S.h.

5.1.2.887 _TIM1_CC1S1

```
#define _TIM1_CC1S1 ((uint8_t) (0x01 << 1))
```

TIM1 Compare 1 selection [1] (in _TIM1_CCMR1)

Definition at line 2958 of file STM8AF_STM8S.h.

5.1.2.888 _TIM1_CC2E

```
#define _TIM1_CC2E ((uint8_t) (0x01 << 4))
```

TIM1 Capture/compare 2 output enable [0] (in _TIM1_CCER1)

Definition at line 3052 of file STM8AF_STM8S.h.

5.1.2.889 _TIM1_CC2G

```
#define _TIM1_CC2G ((uint8_t) (0x01 << 2))
```

TIM1 Capture/compare 2 generation [0] (in _TIM1_EGR)

Definition at line 2948 of file STM8AF_STM8S.h.

5.1.2.890 _TIM1_CC2IE

```
#define _TIM1_CC2IE ((uint8_t) (0x01 << 2))
```

TIM1 Capture/compare 2 interrupt enable [0] (in _TIM1_IER)

Definition at line 2920 of file STM8AF_STM8S.h.

5.1.2.891 _TIM1_CC2IF

```
#define _TIM1_CC2IF ((uint8_t) (0x01 << 2))
```

TIM1 Capture/compare 2 interrupt flag [0] (in _TIM1_SR1)

Definition at line 2930 of file STM8AF_STM8S.h.

5.1.2.892 _TIM1_CC2NE

```
#define _TIM1_CC2NE ((uint8_t) (0x01 << 6))
```

TIM1 Capture/compare 2 complementary output enable [0] (in _TIM1_CCER1)

Definition at line 3054 of file STM8AF_STM8S.h.

5.1.2.893 _TIM1_CC2NP

```
#define _TIM1_CC2NP ((uint8_t) (0x01 << 7))
```

TIM1 Capture/compare 2 complementary output polarity [0] (in _TIM1_CCER1)

Definition at line 3055 of file STM8AF_STM8S.h.

5.1.2.894 _TIM1_CC2OF

```
#define _TIM1_CC2OF ((uint8_t) (0x01 << 2))
```

TIM1 Capture/compare 2 overcapture flag [0] (in _TIM1_SR2)

Definition at line 2940 of file STM8AF_STM8S.h.

5.1.2.895 _TIM1_CC2P

```
#define _TIM1_CC2P ((uint8_t) (0x01 << 5))
```

TIM1 Capture/compare 2 output polarity [0] (in _TIM1_CCER1)

Definition at line 3053 of file STM8AF_STM8S.h.

5.1.2.896 _TIM1_CC2S

```
#define _TIM1_CC2S ((uint8_t) (0x03 << 0))
```

TIM1 Compare 2 selection [1:0] (in _TIM1_CCMR2)

Definition at line 2979 of file STM8AF_STM8S.h.

5.1.2.897 _TIM1_CC2S0

```
#define _TIM1_CC2S0 ((uint8_t) (0x01 << 0))
```

TIM1 Compare 2 selection [0] (in _TIM1_CCMR2)

Definition at line 2980 of file STM8AF_STM8S.h.

5.1.2.898 _TIM1_CC2S1

```
#define _TIM1_CC2S1 ((uint8_t) (0x01 << 1))
```

TIM1 Compare 2 selection [1] (in _TIM1_CCMR2)

Definition at line 2981 of file STM8AF_STM8S.h.

5.1.2.899 _TIM1_CC3E

```
#define _TIM1_CC3E ((uint8_t) (0x01 << 0))
```

TIM1 Capture/compare 3 output enable [0] (in _TIM1_CCER2)

Definition at line 3058 of file STM8AF_STM8S.h.

5.1.2.900 _TIM1_CC3G

```
#define _TIM1_CC3G ((uint8_t) (0x01 << 3))
```

TIM1 Capture/compare 3 generation [0] (in _TIM1_EGR)

Definition at line 2949 of file STM8AF_STM8S.h.

5.1.2.901 _TIM1_CC3IE

```
#define _TIM1_CC3IE ((uint8_t) (0x01 << 3))
```

TIM1 Capture/compare 3 interrupt enable [0] (in _TIM1_IER)

Definition at line 2921 of file STM8AF_STM8S.h.

5.1.2.902 _TIM1_CC3IF

```
#define _TIM1_CC3IF ((uint8_t) (0x01 << 3))
```

TIM1 Capture/compare 3 interrupt flag [0] (in _TIM1_SR1)

Definition at line 2931 of file STM8AF_STM8S.h.

5.1.2.903 _TIM1_CC3NE

```
#define _TIM1_CC3NE ((uint8_t) (0x01 << 2))
```

TIM1 Capture/compare 3 complementary output enable [0] (in _TIM1_CCER2)

Definition at line 3060 of file STM8AF_STM8S.h.

5.1.2.904 _TIM1_CC3NP

```
#define _TIM1_CC3NP ((uint8_t) (0x01 << 3))
```

TIM1 Capture/compare 3 complementary output polarity [0] (in _TIM1_CCER2)

Definition at line 3061 of file STM8AF_STM8S.h.

5.1.2.905 _TIM1_CC3OF

```
#define _TIM1_CC3OF ((uint8_t) (0x01 << 3))
```

TIM1 Capture/compare 3 overcapture flag [0] (in _TIM1_SR2)

Definition at line 2941 of file STM8AF_STM8S.h.

5.1.2.906 _TIM1_CC3P

```
#define _TIM1_CC3P ((uint8_t) (0x01 << 1))
```

TIM1 Capture/compare 3 output polarity [0] (in _TIM1_CCER2)

Definition at line 3059 of file STM8AF_STM8S.h.

5.1.2.907 _TIM1_CC3S

```
#define _TIM1_CC3S ((uint8_t) (0x03 << 0))
```

TIM1 Compare 3 selection [1:0] (in _TIM1_CCMR3)

Definition at line 3002 of file STM8AF_STM8S.h.

5.1.2.908 _TIM1_CC3S0

```
#define _TIM1_CC3S0 ((uint8_t) (0x01 << 0))
```

TIM1 Compare 3 selection [0] (in _TIM1_CCMR3)

Definition at line 3003 of file STM8AF_STM8S.h.

5.1.2.909 _TIM1_CC3S1

```
#define _TIM1_CC3S1 ((uint8_t) (0x01 << 1))
```

TIM1 Compare 3 selection [1] (in _TIM1_CCMR3)

Definition at line 3004 of file STM8AF_STM8S.h.

5.1.2.910 _TIM1_CC4E

```
#define _TIM1_CC4E ((uint8_t) (0x01 << 4))
```

TIM1 Capture/compare 4 output enable [0] (in _TIM1_CCER2)

Definition at line 3062 of file STM8AF_STM8S.h.

5.1.2.911 _TIM1_CC4G

```
#define _TIM1_CC4G ((uint8_t) (0x01 << 4))
```

TIM1 Capture/compare 4 generation [0] (in _TIM1_EGR)

Definition at line 2950 of file STM8AF_STM8S.h.

5.1.2.912 _TIM1_CC4IE

```
#define _TIM1_CC4IE ((uint8_t) (0x01 << 4))
```

TIM1 Capture/compare 4 interrupt enable [0] (in _TIM1_IER)

Definition at line 2922 of file STM8AF_STM8S.h.

5.1.2.913 _TIM1_CC4IF

```
#define _TIM1_CC4IF ((uint8_t) (0x01 << 4))
```

TIM1 Capture/compare 4 interrupt flag [0] (in _TIM1_SR1)

Definition at line 2932 of file STM8AF_STM8S.h.

5.1.2.914 _TIM1_CC4OF

```
#define _TIM1_CC4OF ((uint8_t) (0x01 << 4))
```

TIM1 Capture/compare 4 overcapture flag [0] (in _TIM1_SR2)

Definition at line 2942 of file STM8AF_STM8S.h.

5.1.2.915 _TIM1_CC4P

```
#define _TIM1_CC4P ((uint8_t) (0x01 << 5))
```

TIM1 Capture/compare 4 output polarity [0] (in _TIM1_CCER2)

Definition at line 3063 of file STM8AF_STM8S.h.

5.1.2.916 _TIM1_CC4S

```
#define _TIM1_CC4S ((uint8_t) (0x03 << 0))
```

TIM1 Compare 4 selection [1:0] (in _TIM1_CCMR4)

Definition at line 3025 of file STM8AF_STM8S.h.

5.1.2.917 _TIM1_CC4S0

```
#define _TIM1_CC4S0 ((uint8_t) (0x01 << 0))
```

TIM1 Compare 4 selection [0] (in _TIM1_CCMR4)

Definition at line 3026 of file STM8AF_STM8S.h.

5.1.2.918 _TIM1_CC4S1

```
#define _TIM1_CC4S1 ((uint8_t) (0x01 << 1))
```

TIM1 Compare 4 selection [1] (in _TIM1_CCMR4)

Definition at line 3027 of file STM8AF_STM8S.h.

5.1.2.919 _TIM1_CCER1

```
#define _TIM1_CCER1 _SFR(uint8_t, TIM1_AddressBase+0x0C)
```

TIM1 Capture/compare enable register 1.

Definition at line 2816 of file STM8AF_STM8S.h.

5.1.2.920 _TIM1_CCER1_RESET_VALUE

```
#define _TIM1_CCER1_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 Capture/compare enable register 1 reset value.

Definition at line 2850 of file STM8AF_STM8S.h.

5.1.2.921 _TIM1_CCER2

```
#define _TIM1_CCER2 _SFR(uint8_t, TIM1_AddressBase+0x0D)
```

TIM1 Capture/compare enable register 2.

Definition at line 2817 of file STM8AF_STM8S.h.

5.1.2.922 _TIM1_CCER2_RESET_VALUE

```
#define _TIM1_CCER2_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 Capture/compare enable register 2 reset value.

Definition at line 2851 of file STM8AF_STM8S.h.

5.1.2.923 _TIM1_CCMR1

```
#define _TIM1_CCMR1 _SFR(uint8_t, TIM1_AddressBase+0x08)
```

TIM1 Capture/compare mode register 1.

Definition at line 2812 of file STM8AF_STM8S.h.

5.1.2.924 _TIM1_CCMR1_RESET_VALUE

```
#define _TIM1_CCMR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 Capture/compare mode register 1 reset value.

Definition at line 2846 of file STM8AF_STM8S.h.

5.1.2.925 _TIM1_CCMR2

```
#define _TIM1_CCMR2 _SFR(uint8_t, TIM1_AddressBase+0x09)
```

TIM1 Capture/compare mode register 2.

Definition at line 2813 of file STM8AF_STM8S.h.

5.1.2.926 _TIM1_CCMR2_RESET_VALUE

```
#define _TIM1_CCMR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 Capture/compare mode register 2 reset value.

Definition at line 2847 of file STM8AF_STM8S.h.

5.1.2.927 _TIM1_CCMR3

```
#define _TIM1_CCMR3 _SFR(uint8_t, TIM1_AddressBase+0x0A)
```

TIM1 Capture/compare mode register 3.

Definition at line 2814 of file STM8AF_STM8S.h.

5.1.2.928 _TIM1_CCMR3_RESET_VALUE

```
#define _TIM1_CCMR3_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 Capture/compare mode register 3 reset value.

Definition at line 2848 of file STM8AF_STM8S.h.

5.1.2.929 _TIM1_CCMR4

```
#define _TIM1_CCMR4 _SFR(uint8_t, TIM1_AddressBase+0x0B)
```

TIM1 Capture/compare mode register 4.

Definition at line 2815 of file STM8AF_STM8S.h.

5.1.2.930 _TIM1_CCMR4_RESET_VALUE

```
#define _TIM1_CCMR4_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 Capture/compare mode register 4 reset value.

Definition at line 2849 of file STM8AF_STM8S.h.

5.1.2.931 _TIM1_CCPC

```
#define _TIM1_CCPC ((uint8_t) (0x01 << 0))
```

TIM1 Capture/compare preloaded control [0] (in _TIM1_CR2)

Definition at line 2883 of file STM8AF_STM8S.h.

5.1.2.932 _TIM1_CCR1H

```
#define _TIM1_CCR1H _SFR(uint8_t, TIM1_AddressBase+0x15)
```

TIM1 16-bit capture/compare value 1 high byte.

Definition at line 2825 of file STM8AF_STM8S.h.

5.1.2.933 _TIM1_CCR1H_RESET_VALUE

```
#define _TIM1_CCR1H_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 16-bit capture/compare value 1 high byte reset value.

Definition at line 2859 of file STM8AF_STM8S.h.

5.1.2.934 _TIM1_CCR1L

```
#define _TIM1_CCR1L _SFR(uint8_t, TIM1_AddressBase+0x16)
```

TIM1 16-bit capture/compare value 1 low byte.

Definition at line 2826 of file STM8AF_STM8S.h.

5.1.2.935 _TIM1_CCR1L_RESET_VALUE

```
#define _TIM1_CCR1L_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 16-bit capture/compare value 1 low byte reset value.

Definition at line 2860 of file STM8AF_STM8S.h.

5.1.2.936 _TIM1_CCR2H

```
#define _TIM1_CCR2H _SFR(uint8_t, TIM1_AddressBase+0x17)
```

TIM1 16-bit capture/compare value 2 high byte.

Definition at line 2827 of file STM8AF_STM8S.h.

5.1.2.937 _TIM1_CCR2H_RESET_VALUE

```
#define _TIM1_CCR2H_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 16-bit capture/compare value 2 high byte reset value.

Definition at line 2861 of file STM8AF_STM8S.h.

5.1.2.938 _TIM1_CCR2L

```
#define _TIM1_CCR2L _SFR(uint8_t, TIM1_AddressBase+0x18)
```

TIM1 16-bit capture/compare value 2 low byte.

Definition at line 2828 of file STM8AF_STM8S.h.

5.1.2.939 _TIM1_CCR2L_RESET_VALUE

```
#define _TIM1_CCR2L_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 16-bit capture/compare value 2 low byte reset value.

Definition at line 2862 of file STM8AF_STM8S.h.

5.1.2.940 _TIM1_CCR3H

```
#define _TIM1_CCR3H _SFR(uint8_t, TIM1_AddressBase+0x19)
```

TIM1 16-bit capture/compare value 3 high byte.

Definition at line 2829 of file STM8AF_STM8S.h.

5.1.2.941 _TIM1_CCR3H_RESET_VALUE

```
#define _TIM1_CCR3H_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 16-bit capture/compare value 3 high byte reset value.

Definition at line 2863 of file STM8AF_STM8S.h.

5.1.2.942 _TIM1_CCR3L

```
#define _TIM1_CCR3L _SFR(uint8_t, TIM1_AddressBase+0x1A)
```

TIM1 16-bit capture/compare value 3 low byte.

Definition at line 2830 of file STM8AF_STM8S.h.

5.1.2.943 _TIM1_CCR3L_RESET_VALUE

```
#define _TIM1_CCR3L_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 16-bit capture/compare value 3 low byte reset value.

Definition at line 2864 of file STM8AF_STM8S.h.

5.1.2.944 _TIM1_CCR4H

```
#define _TIM1_CCR4H _SFR(uint8_t, TIM1_AddressBase+0x1B)
```

TIM1 16-bit capture/compare value 4 high byte.

Definition at line 2831 of file STM8AF_STM8S.h.

5.1.2.945 _TIM1_CCR4H_RESET_VALUE

```
#define _TIM1_CCR4H_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 16-bit capture/compare value 4 high byte reset value.

Definition at line 2865 of file STM8AF_STM8S.h.

5.1.2.946 _TIM1_CCR4L

```
#define _TIM1_CCR4L _SFR(uint8_t, TIM1_AddressBase+0x1C)
```

TIM1 16-bit capture/compare value 4 low byte.

Definition at line 2832 of file STM8AF_STM8S.h.

5.1.2.947 _TIM1_CCR4L_RESET_VALUE

```
#define _TIM1_CCR4L_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 16-bit capture/compare value 4 low byte reset value.

Definition at line 2866 of file STM8AF_STM8S.h.

5.1.2.948 _TIM1_CEN

```
#define _TIM1_CEN ((uint8_t) (0x01 << 0))
```

TIM1 Counter enable [0] (in _TIM1_CR1)

Definition at line 2872 of file STM8AF_STM8S.h.

5.1.2.949 _TIM1_CMS

```
#define _TIM1_CMS ((uint8_t) (0x03 << 5))
```

TIM1 Center-aligned mode selection [1:0] (in _TIM1_CR1)

Definition at line 2877 of file STM8AF_STM8S.h.

5.1.2.950 _TIM1_CMS0

```
#define _TIM1_CMS0 ((uint8_t) (0x01 << 5))
```

TIM1 Center-aligned mode selection [0] (in _TIM1_CR1)

Definition at line 2878 of file STM8AF_STM8S.h.

5.1.2.951 _TIM1_CMS1

```
#define _TIM1_CMS1 ((uint8_t) (0x01 << 6))
```

TIM1 Center-aligned mode selection [1] (in _TIM1_CR1)

Definition at line 2879 of file STM8AF_STM8S.h.

5.1.2.952 _TIM1_CNTRH

```
#define _TIM1_CNTRH _SFR(uint8_t, TIM1_AddressBase+0x0E)
```

TIM1 counter register high byte.

Definition at line 2818 of file STM8AF_STM8S.h.

5.1.2.953 _TIM1_CNTRH_RESET_VALUE

```
#define _TIM1_CNTRH_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 counter register high byte reset value.

Definition at line 2852 of file STM8AF_STM8S.h.

5.1.2.954 _TIM1_CNTRL

```
#define _TIM1_CNTRL _SFR(uint8_t, TIM1_AddressBase+0x0F)
```

TIM1 counter register low byte.

Definition at line 2819 of file STM8AF_STM8S.h.

5.1.2.955 _TIM1_CNTRL_RESET_VALUE

```
#define _TIM1_CNTRL_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 counter register low byte reset value.

Definition at line 2853 of file STM8AF_STM8S.h.

5.1.2.956 _TIM1_COMG

```
#define _TIM1_COMG ((uint8_t) (0x01 << 5))
```

TIM1 Capture/compare control update generation [0] (in _TIM1_EGR)

Definition at line 2951 of file STM8AF_STM8S.h.

5.1.2.957 _TIM1_COMIE

```
#define _TIM1_COMIE ((uint8_t) (0x01 << 5))
```

TIM1 Commutation interrupt enable [0] (in _TIM1_IER)

Definition at line 2923 of file STM8AF_STM8S.h.

5.1.2.958 _TIM1_COMIF

```
#define _TIM1_COMIF ((uint8_t) (0x01 << 5))
```

TIM1 Commutation interrupt flag [0] (in _TIM1_SR1)

Definition at line 2933 of file STM8AF_STM8S.h.

5.1.2.959 _TIM1_COMS

```
#define _TIM1_COMS ((uint8_t) (0x01 << 2))
```

TIM1 Capture/compare control update selection [0] (in _TIM1_CR2)

Definition at line 2885 of file STM8AF_STM8S.h.

5.1.2.960 _TIM1_CR1

```
#define _TIM1_CR1 _SFR(uint8_t, TIM1_AddressBase+0x00)
```

TIM1 control register 1.

Definition at line 2804 of file STM8AF_STM8S.h.

5.1.2.961 _TIM1_CR1_RESET_VALUE

```
#define _TIM1_CR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 control register 1 reset value.

Definition at line 2838 of file STM8AF_STM8S.h.

5.1.2.962 _TIM1_CR2

```
#define _TIM1_CR2 _SFR(uint8_t, TIM1_AddressBase+0x01)
```

TIM1 control register 2.

Definition at line 2805 of file STM8AF_STM8S.h.

5.1.2.963 _TIM1_CR2_RESET_VALUE

```
#define _TIM1_CR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 control register 2 reset value.

Definition at line 2839 of file STM8AF_STM8S.h.

5.1.2.964 _TIM1_DIR

```
#define _TIM1_DIR ((uint8_t) (0x01 << 4))
```

TIM1 Direction [0] (in _TIM1_CR1)

Definition at line 2876 of file STM8AF_STM8S.h.

5.1.2.965 _TIM1_DTR

```
#define _TIM1_DTR _SFR(uint8_t, TIM1_AddressBase+0x1E)
```

TIM1 Dead-time register.

Definition at line 2834 of file STM8AF_STM8S.h.

5.1.2.966 _TIM1_DTR_RESET_VALUE

```
#define _TIM1_DTR_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 Dead-time register reset value.

Definition at line 2868 of file STM8AF_STM8S.h.

5.1.2.967 _TIM1_ECE

```
#define _TIM1_ECE ((uint8_t) (0x01 << 6))
```

TIM1 External clock enable [0] (in _TIM1_ETR)

Definition at line 2914 of file STM8AF_STM8S.h.

5.1.2.968 _TIM1_EGR

```
#define _TIM1_EGR _SFR(uint8_t, TIM1_AddressBase+0x07)
```

TIM1 Event generation register.

Definition at line 2811 of file STM8AF_STM8S.h.

5.1.2.969 _TIM1_EGR_RESET_VALUE

```
#define _TIM1_EGR_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 Event generation register reset value.

Definition at line 2845 of file STM8AF_STM8S.h.

5.1.2.970 _TIM1_ETF

```
#define _TIM1_ETF ((uint8_t) (0x0F << 0))
```

TIM1 External trigger filter [3:0] (in _TIM1_ETR)

Definition at line 2906 of file STM8AF_STM8S.h.

5.1.2.971 _TIM1_ETF0

```
#define _TIM1_ETF0 ((uint8_t) (0x01 << 0))
```

TIM1 External trigger filter [0] (in _TIM1_ETR)

Definition at line 2907 of file STM8AF_STM8S.h.

5.1.2.972 _TIM1_ETF1

```
#define _TIM1_ETF1 ((uint8_t) (0x01 << 1))
```

TIM1 External trigger filter [1] (in _TIM1_ETR)

Definition at line 2908 of file STM8AF_STM8S.h.

5.1.2.973 _TIM1_ETF2

```
#define _TIM1_ETF2 ((uint8_t) (0x01 << 2))
```

TIM1 External trigger filter [2] (in _TIM1_ETR)

Definition at line 2909 of file STM8AF_STM8S.h.

5.1.2.974 _TIM1_ETF3

```
#define _TIM1_ETF3 ((uint8_t) (0x01 << 3))
```

TIM1 External trigger filter [3] (in _TIM1_ETR)

Definition at line 2910 of file STM8AF_STM8S.h.

5.1.2.975 _TIM1_ETP

```
#define _TIM1_ETP ((uint8_t) (0x01 << 7))
```

TIM1 External trigger polarity [0] (in _TIM1_ETR)

Definition at line 2915 of file STM8AF_STM8S.h.

5.1.2.976 _TIM1_ETPS

```
#define _TIM1_ETPS ((uint8_t) (0x03 << 4))
```

TIM1 External trigger prescaler [1:0] (in _TIM1_ETR)

Definition at line 2911 of file STM8AF_STM8S.h.

5.1.2.977 _TIM1_ETPS0

```
#define _TIM1_ETPS0 ((uint8_t) (0x01 << 4))
```

TIM1 External trigger prescaler [0] (in _TIM1_ETR)

Definition at line 2912 of file STM8AF_STM8S.h.

5.1.2.978 _TIM1_ETPS1

```
#define _TIM1_ETPS1 ((uint8_t) (0x01 << 5))
```

TIM1 External trigger prescaler [1] (in _TIM1_ETR)

Definition at line 2913 of file STM8AF_STM8S.h.

5.1.2.979 _TIM1_ETR

```
#define _TIM1_ETR _SFR(uint8_t, TIM1_AddressBase+0x03)
```

TIM1 External trigger register.

Definition at line 2807 of file STM8AF_STM8S.h.

5.1.2.980 _TIM1_ETR_RESET_VALUE

```
#define _TIM1_ETR_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 External trigger register reset value.

Definition at line 2841 of file STM8AF_STM8S.h.

5.1.2.981 _TIM1_IC1F

```
#define _TIM1_IC1F ((uint8_t) (0x0F << 4))
```

TIM1 Output compare 1 mode [3:0] (in _TIM1_CCMR1)

Definition at line 2972 of file STM8AF_STM8S.h.

5.1.2.982 _TIM1_IC1F0

```
#define _TIM1_IC1F0 ((uint8_t) (0x01 << 4))
```

TIM1 Input capture 1 filter [0] (in _TIM1_CCMR1)

Definition at line 2973 of file STM8AF_STM8S.h.

5.1.2.983 _TIM1_IC1F1

```
#define _TIM1_IC1F1 ((uint8_t) (0x01 << 5))
```

TIM1 Input capture 1 filter [1] (in _TIM1_CCMR1)

Definition at line 2974 of file STM8AF_STM8S.h.

5.1.2.984 _TIM1_IC1F2

```
#define _TIM1_IC1F2 ((uint8_t) (0x01 << 6))
```

TIM1 Input capture 1 filter [2] (in _TIM1_CCMR1)

Definition at line 2975 of file STM8AF_STM8S.h.

5.1.2.985 _TIM1_IC1F3

```
#define _TIM1_IC1F3 ((uint8_t) (0x01 << 7))
```

TIM1 Input capture 1 filter [3] (in _TIM1_CCMR1)

Definition at line 2976 of file STM8AF_STM8S.h.

5.1.2.986 _TIM1_IC1PSC

```
#define _TIM1_IC1PSC ((uint8_t) (0x03 << 2))
```

TIM1 Input capture 1 prescaler [1:0] (in _TIM1_CCMR1)

Definition at line 2969 of file STM8AF_STM8S.h.

5.1.2.987 _TIM1_IC1PSC0

```
#define _TIM1_IC1PSC0 ((uint8_t) (0x01 << 2))
```

TIM1 Input capture 1 prescaler [0] (in _TIM1_CCMR1)

Definition at line 2970 of file STM8AF_STM8S.h.

5.1.2.988 _TIM1_IC1PSC1

```
#define _TIM1_IC1PSC1 ((uint8_t) (0x01 << 3))
```

TIM1 Input capture 1 prescaler [1] (in _TIM1_CCMR1)

Definition at line 2971 of file STM8AF_STM8S.h.

5.1.2.989 _TIM1_IC2F

```
#define _TIM1_IC2F ((uint8_t) (0x0F << 4))
```

TIM1 Output compare 2 mode [3:0] (in _TIM1_CCMR2)

Definition at line 2995 of file STM8AF_STM8S.h.

5.1.2.990 _TIM1_IC2F0

```
#define _TIM1_IC2F0 ((uint8_t) (0x01 << 4))
```

TIM1 Input capture 2 filter [0] (in _TIM1_CCMR2)

Definition at line 2996 of file STM8AF_STM8S.h.

5.1.2.991 _TIM1_IC2F1

```
#define _TIM1_IC2F1 ((uint8_t) (0x01 << 5))
```

TIM1 Input capture 2 filter [1] (in _TIM1_CCMR2)

Definition at line 2997 of file STM8AF_STM8S.h.

5.1.2.992 _TIM1_IC2F2

```
#define _TIM1_IC2F2 ((uint8_t) (0x01 << 6))
```

TIM1 Input capture 2 filter [2] (in _TIM1_CCMR2)

Definition at line 2998 of file STM8AF_STM8S.h.

5.1.2.993 _TIM1_IC2F3

```
#define _TIM1_IC2F3 ((uint8_t) (0x01 << 7))
```

TIM1 Input capture 2 filter [3] (in _TIM1_CCMR2)

Definition at line 2999 of file STM8AF_STM8S.h.

5.1.2.994 _TIM1_IC2PSC

```
#define _TIM1_IC2PSC ((uint8_t) (0x03 << 2))
```

TIM1 Input capture 2 prescaler [1:0] (in _TIM1_CCMR2)

Definition at line 2992 of file STM8AF_STM8S.h.

5.1.2.995 _TIM1_IC2PSC0

```
#define _TIM1_IC2PSC0 ((uint8_t) (0x01 << 2))
```

TIM1 Input capture 2 prescaler [0] (in _TIM1_CCMR2)

Definition at line 2993 of file STM8AF_STM8S.h.

5.1.2.996 _TIM1_IC2PSC1

```
#define _TIM1_IC2PSC1 ((uint8_t) (0x01 << 3))
```

TIM1 Input capture 2 prescaler [1] (in _TIM1_CCMR2)

Definition at line 2994 of file STM8AF_STM8S.h.

5.1.2.997 _TIM1_IC3F

```
#define _TIM1_IC3F ((uint8_t) (0x0F << 4))
```

TIM1 Output compare 3 mode [3:0] (in _TIM1_CCMR3)

Definition at line 3018 of file STM8AF_STM8S.h.

5.1.2.998 _TIM1_IC3F0

```
#define _TIM1_IC3F0 ((uint8_t) (0x01 << 4))
```

TIM1 Input capture 3 filter [0] (in _TIM1_CCMR3)

Definition at line 3019 of file STM8AF_STM8S.h.

5.1.2.999 _TIM1_IC3F1

```
#define _TIM1_IC3F1 ((uint8_t) (0x01 << 5))
```

TIM1 Input capture 3 filter [1] (in _TIM1_CCMR3)

Definition at line 3020 of file STM8AF_STM8S.h.

5.1.2.1000 _TIM1_IC3F2

```
#define _TIM1_IC3F2 ((uint8_t) (0x01 << 6))
```

TIM1 Input capture 3 filter [2] (in _TIM1_CCMR3)

Definition at line 3021 of file STM8AF_STM8S.h.

5.1.2.1001 _TIM1_IC3F3

```
#define _TIM1_IC3F3 ((uint8_t) (0x01 << 7))
```

TIM1 Input capture 3 filter [3] (in _TIM1_CCMR3)

Definition at line 3022 of file STM8AF_STM8S.h.

5.1.2.1002 _TIM1_IC3PSC

```
#define _TIM1_IC3PSC ((uint8_t) (0x03 << 2))
```

TIM1 Input capture 3 prescaler [1:0] (in _TIM1_CCMR3)

Definition at line 3015 of file STM8AF_STM8S.h.

5.1.2.1003 _TIM1_IC3PSC0

```
#define _TIM1_IC3PSC0 ((uint8_t) (0x01 << 2))
```

TIM1 Input capture 3 prescaler [0] (in _TIM1_CCMR3)

Definition at line 3016 of file STM8AF_STM8S.h.

5.1.2.1004 _TIM1_IC3PSC1

```
#define _TIM1_IC3PSC1 ((uint8_t) (0x01 << 3))
```

TIM1 Input capture 3 prescaler [1] (in _TIM1_CCMR3)

Definition at line 3017 of file STM8AF_STM8S.h.

5.1.2.1005 _TIM1_IC4F

```
#define _TIM1_IC4F ((uint8_t) (0x0F << 4))
```

TIM1 Output compare 4 mode [3:0] (in _TIM1_CCMR4)

Definition at line 3041 of file STM8AF_STM8S.h.

5.1.2.1006 _TIM1_IC4F0

```
#define _TIM1_IC4F0 ((uint8_t) (0x01 << 4))
```

TIM1 Input capture 4 filter [0] (in _TIM1_CCMR4)

Definition at line 3042 of file STM8AF_STM8S.h.

5.1.2.1007 _TIM1_IC4F1

```
#define _TIM1_IC4F1 ((uint8_t) (0x01 << 5))
```

TIM1 Input capture 4 filter [1] (in _TIM1_CCMR4)

Definition at line 3043 of file STM8AF_STM8S.h.

5.1.2.1008 _TIM1_IC4F2

```
#define _TIM1_IC4F2 ((uint8_t) (0x01 << 6))
```

TIM1 Input capture 4 filter [2] (in _TIM1_CCMR4)

Definition at line 3044 of file STM8AF_STM8S.h.

5.1.2.1009 _TIM1_IC4F3

```
#define _TIM1_IC4F3 ((uint8_t) (0x01 << 7))
```

TIM1 Input capture 4 filter [3] (in _TIM1_CCMR4)

Definition at line 3045 of file STM8AF_STM8S.h.

5.1.2.1010 _TIM1_IC4PSC

```
#define _TIM1_IC4PSC ((uint8_t) (0x03 << 2))
```

TIM1 Input capture 4 prescaler [1:0] (in _TIM1_CCMR4)

Definition at line 3038 of file STM8AF_STM8S.h.

5.1.2.1011 _TIM1_IC4PSC0

```
#define _TIM1_IC4PSC0 ((uint8_t) (0x01 << 2))
```

TIM1 Input capture 4 prescaler [0] (in _TIM1_CCMR4)

Definition at line 3039 of file STM8AF_STM8S.h.

5.1.2.1012 _TIM1_IC4PSC1

```
#define _TIM1_IC4PSC1 ((uint8_t) (0x01 << 3))
```

TIM1 Input capture 4 prescaler [1] (in _TIM1_CCMR4)

Definition at line 3040 of file STM8AF_STM8S.h.

5.1.2.1013 _TIM1_IER

```
#define _TIM1_IER _SFR(uint8_t, TIM1_AddressBase+0x04)
```

TIM1 interrupt enable register.

Definition at line 2808 of file STM8AF_STM8S.h.

5.1.2.1014 _TIM1_IER_RESET_VALUE

```
#define _TIM1_IER_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 interrupt enable register reset value.

Definition at line 2842 of file STM8AF_STM8S.h.

5.1.2.1015 _TIM1_LOCK

```
#define _TIM1_LOCK ((int8_t) (0x03 << 0))
```

TIM1 Lock configuration [1:0] (in _TIM1_BKR)

Definition at line 3067 of file STM8AF_STM8S.h.

5.1.2.1016 _TIM1_LOCK0

```
#define _TIM1_LOCK0 ((uint8_t) (0x01 << 0))
```

TIM1 Lock configuration [0] (in _TIM1_BKR)

Definition at line 3068 of file STM8AF_STM8S.h.

5.1.2.1017 _TIM1_LOCK1

```
#define _TIM1_LOCK1 ((uint8_t) (0x01 << 1))
```

TIM1 Lock configuration [1] (in _TIM1_BKR)

Definition at line 3069 of file STM8AF_STM8S.h.

5.1.2.1018 _TIM1_MMS

```
#define _TIM1_MMS ((uint8_t) (0x07 << 4))
```

TIM1 Master mode selection [2:0] (in _TIM1_CR2)

Definition at line 2887 of file STM8AF_STM8S.h.

5.1.2.1019 _TIM1_MMS0

```
#define _TIM1_MMS0 ((uint8_t) (0x01 << 4))
```

TIM1 Master mode selection [0] (in _TIM1_CR2)

Definition at line 2888 of file STM8AF_STM8S.h.

5.1.2.1020 _TIM1_MMS1

```
#define _TIM1_MMS1 ((uint8_t) (0x01 << 5))
```

TIM1 Master mode selection [1] (in _TIM1_CR2)

Definition at line 2889 of file STM8AF_STM8S.h.

5.1.2.1021 _TIM1_MMS2

```
#define _TIM1_MMS2 ((uint8_t) (0x01 << 6))
```

TIM1 Master mode selection [2] (in _TIM1_CR2)

Definition at line 2890 of file STM8AF_STM8S.h.

5.1.2.1022 _TIM1_MOE

```
#define _TIM1_MOE ((uint8_t) (0x01 << 7))
```

TIM1 Main output enable [0] (in _TIM1_BKR)

Definition at line 3075 of file STM8AF_STM8S.h.

5.1.2.1023 _TIM1_MSM

```
#define _TIM1_MSM ((uint8_t) (0x01 << 7))
```

TIM1 Master/slave mode [0] (in _TIM1_SMCR)

Definition at line 2903 of file STM8AF_STM8S.h.

5.1.2.1024 _TIM1_OC1CE

```
#define _TIM1_OC1CE ((uint8_t) (0x01 << 7))
```

TIM1 Output compare 1 clear enable [0] (in _TIM1_CCMR1)

Definition at line 2965 of file STM8AF_STM8S.h.

5.1.2.1025 _TIM1_OC1FE

```
#define _TIM1_OC1FE ((uint8_t) (0x01 << 2))
```

TIM1 Output compare 1 fast enable [0] (in _TIM1_CCMR1)

Definition at line 2959 of file STM8AF_STM8S.h.

5.1.2.1026 _TIM1_OC1M

```
#define _TIM1_OC1M ((uint8_t) (0x07 << 4))
```

TIM1 Output compare 1 mode [2:0] (in _TIM1_CCMR1)

Definition at line 2961 of file STM8AF_STM8S.h.

5.1.2.1027 _TIM1_OC1M0

```
#define _TIM1_OC1M0 ((uint8_t) (0x01 << 4))
```

TIM1 Output compare 1 mode [0] (in _TIM1_CCMR1)

Definition at line 2962 of file STM8AF_STM8S.h.

5.1.2.1028 _TIM1_OC1M1

```
#define _TIM1_OC1M1 ((uint8_t) (0x01 << 5))
```

TIM1 Output compare 1 mode [1] (in _TIM1_CCMR1)

Definition at line 2963 of file STM8AF_STM8S.h.

5.1.2.1029 _TIM1_OC1M2

```
#define _TIM1_OC1M2 ((uint8_t) (0x01 << 6))
```

TIM1 Output compare 1 mode [2] (in _TIM1_CCMR1)

Definition at line 2964 of file STM8AF_STM8S.h.

5.1.2.1030 _TIM1_OC1PE

```
#define _TIM1_OC1PE ((uint8_t) (0x01 << 3))
```

TIM1 Output compare 1 preload enable [0] (in _TIM1_CCMR1)

Definition at line 2960 of file STM8AF_STM8S.h.

5.1.2.1031 _TIM1_OC2CE

```
#define _TIM1_OC2CE ((uint8_t) (0x01 << 7))
```

TIM1 Output compare 2 clear enable [0] (in _TIM1_CCMR2)

Definition at line 2988 of file STM8AF_STM8S.h.

5.1.2.1032 _TIM1_OC2FE

```
#define _TIM1_OC2FE ((uint8_t) (0x01 << 2))
```

TIM1 Output compare 2 fast enable [0] (in _TIM1_CCMR2)

Definition at line 2982 of file STM8AF_STM8S.h.

5.1.2.1033 _TIM1_OC2M

```
#define _TIM1_OC2M ((uint8_t) (0x07 << 4))
```

TIM1 Output compare 2 mode [2:0] (in _TIM1_CCMR2)

Definition at line 2984 of file STM8AF_STM8S.h.

5.1.2.1034 _TIM1_OC2M0

```
#define _TIM1_OC2M0 ((uint8_t) (0x01 << 4))
```

TIM1 Output compare 2 mode [0] (in _TIM1_CCMR2)

Definition at line 2985 of file STM8AF_STM8S.h.

5.1.2.1035 _TIM1_OC2M1

```
#define _TIM1_OC2M1 ((uint8_t) (0x01 << 5))
```

TIM1 Output compare 2 mode [1] (in _TIM1_CCMR2)

Definition at line 2986 of file STM8AF_STM8S.h.

5.1.2.1036 _TIM1_OC2M2

```
#define _TIM1_OC2M2 ((uint8_t) (0x01 << 6))
```

TIM1 Output compare 2 mode [2] (in _TIM1_CCMR2)

Definition at line 2987 of file STM8AF_STM8S.h.

5.1.2.1037 _TIM1_OC2PE

```
#define _TIM1_OC2PE ((uint8_t) (0x01 << 3))
```

TIM1 Output compare 2 preload enable [0] (in _TIM1_CCMR2)

Definition at line 2983 of file STM8AF_STM8S.h.

5.1.2.1038 _TIM1_OC3CE

```
#define _TIM1_OC3CE ((uint8_t) (0x01 << 7))
```

TIM1 Output compare 3 clear enable [0] (in _TIM1_CCMR3)

Definition at line 3011 of file STM8AF_STM8S.h.

5.1.2.1039 _TIM1_OC3FE

```
#define _TIM1_OC3FE ((uint8_t) (0x01 << 2))
```

TIM1 Output compare 3 fast enable [0] (in _TIM1_CCMR3)

Definition at line 3005 of file STM8AF_STM8S.h.

5.1.2.1040 _TIM1_OC3M

```
#define _TIM1_OC3M ((uint8_t) (0x07 << 4))
```

TIM1 Output compare 3 mode [2:0] (in _TIM1_CCMR3)

Definition at line 3007 of file STM8AF_STM8S.h.

5.1.2.1041 _TIM1_OC3M0

```
#define _TIM1_OC3M0 ((uint8_t) (0x01 << 4))
```

TIM1 Output compare 3 mode [0] (in _TIM1_CCMR3)

Definition at line 3008 of file STM8AF_STM8S.h.

5.1.2.1042 _TIM1_OC3M1

```
#define _TIM1_OC3M1 ((uint8_t) (0x01 << 5))
```

TIM1 Output compare 3 mode [1] (in _TIM1_CCMR3)

Definition at line 3009 of file STM8AF_STM8S.h.

5.1.2.1043 _TIM1_OC3M2

```
#define _TIM1_OC3M2 ((uint8_t) (0x01 << 6))
```

TIM1 Output compare 3 mode [2] (in _TIM1_CCMR3)

Definition at line 3010 of file STM8AF_STM8S.h.

5.1.2.1044 _TIM1_OC3PE

```
#define _TIM1_OC3PE ((uint8_t) (0x01 << 3))
```

TIM1 Output compare 3 preload enable [0] (in _TIM1_CCMR3)

Definition at line 3006 of file STM8AF_STM8S.h.

5.1.2.1045 _TIM1_OC4CE

```
#define _TIM1_OC4CE ((uint8_t) (0x01 << 7))
```

TIM1 Output compare 4 clear enable [0] (in _TIM1_CCMR4)

Definition at line 3034 of file STM8AF_STM8S.h.

5.1.2.1046 _TIM1_OC4FE

```
#define _TIM1_OC4FE ((uint8_t) (0x01 << 2))
```

TIM1 Output compare 4 fast enable [0] (in _TIM1_CCMR4)

Definition at line 3028 of file STM8AF_STM8S.h.

5.1.2.1047 _TIM1_OC4M

```
#define _TIM1_OC4M ((uint8_t) (0x07 << 4))
```

TIM1 Output compare 4 mode [2:0] (in _TIM1_CCMR4)

Definition at line 3030 of file STM8AF_STM8S.h.

5.1.2.1048 _TIM1_OC4M0

```
#define _TIM1_OC4M0 ((uint8_t) (0x01 << 4))
```

TIM1 Output compare 4 mode [0] (in _TIM1_CCMR4)

Definition at line 3031 of file STM8AF_STM8S.h.

5.1.2.1049 _TIM1_OC4M1

```
#define _TIM1_OC4M1 ((uint8_t) (0x01 << 5))
```

TIM1 Output compare 4 mode [1] (in _TIM1_CCMR4)

Definition at line 3032 of file STM8AF_STM8S.h.

5.1.2.1050 _TIM1_OC4M2

```
#define _TIM1_OC4M2 ((uint8_t) (0x01 << 6))
```

TIM1 Output compare 4 mode [2] (in _TIM1_CCMR4)

Definition at line 3033 of file STM8AF_STM8S.h.

5.1.2.1051 _TIM1_OC4PE

```
#define _TIM1_OC4PE ((uint8_t) (0x01 << 3))
```

TIM1 Output compare 4 preload enable [0] (in _TIM1_CCMR4)

Definition at line 3029 of file STM8AF_STM8S.h.

5.1.2.1052 _TIM1_OIS1

```
#define _TIM1_OIS1 ((uint8_t) (0x01 << 0))
```

TIM1 Output idle state 1 (OC1 output) [0] (in _TIM1_OISR)

Definition at line 3078 of file STM8AF_STM8S.h.

5.1.2.1053 _TIM1_OIS1N

```
#define _TIM1_OIS1N ((uint8_t) (0x01 << 1))
```

TIM1 Output idle state 1 (OC1N output) [0] (in _TIM1_OISR)

Definition at line 3079 of file STM8AF_STM8S.h.

5.1.2.1054 _TIM1_OIS2

```
#define _TIM1_OIS2 ((uint8_t) (0x01 << 2))
```

TIM1 Output idle state 2 (OC2 output) [0] (in _TIM1_OISR)

Definition at line 3080 of file STM8AF_STM8S.h.

5.1.2.1055 _TIM1_OIS2N

```
#define _TIM1_OIS2N ((uint8_t) (0x01 << 3))
```

TIM1 Output idle state 2 (OC2N output) [0] (in _TIM1_OISR)

Definition at line 3081 of file STM8AF_STM8S.h.

5.1.2.1056 _TIM1_OIS3

```
#define _TIM1_OIS3 ((uint8_t) (0x01 << 4))
```

TIM1 Output idle state 3 (OC3 output) [0] (in _TIM1_OISR)

Definition at line 3082 of file STM8AF_STM8S.h.

5.1.2.1057 _TIM1_OIS3N

```
#define _TIM1_OIS3N ((uint8_t) (0x01 << 5))
```

TIM1 Output idle state 3 (OC3N output) [0] (in _TIM1_OISR)

Definition at line 3083 of file STM8AF_STM8S.h.

5.1.2.1058 _TIM1_OIS4

```
#define _TIM1_OIS4 ((uint8_t) (0x01 << 6))
```

TIM1 Output idle state 4 (OC4 output) [0] (in _TIM1_OISR)

Definition at line 3084 of file STM8AF_STM8S.h.

5.1.2.1059 _TIM1_OISR

```
#define _TIM1_OISR _SFR(uint8_t, TIM1_AddressBase+0x1F)
```

TIM1 Output idle state register.

Definition at line 2835 of file STM8AF_STM8S.h.

5.1.2.1060 _TIM1_OISR_RESET_VALUE

```
#define _TIM1_OISR_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 Output idle state register reset value.

Definition at line 2869 of file STM8AF_STM8S.h.

5.1.2.1061 _TIM1_OPM

```
#define _TIM1_OPM ((uint8_t) (0x01 << 3))
```

TIM1 One-pulse mode [0] (in _TIM1_CR1)

Definition at line 2875 of file STM8AF_STM8S.h.

5.1.2.1062 _TIM1_OSSI

```
#define _TIM1_OSSI ((uint8_t) (0x01 << 2))
```

TIM1 Off state selection for idle mode [0] (in _TIM1_BKR)

Definition at line 3070 of file STM8AF_STM8S.h.

5.1.2.1063 _TIM1_OSSR

```
#define _TIM1_OSSR ((uint8_t) (0x01 << 3))
```

TIM1 Off state selection for Run mode [0] (in _TIM1_BKR)

Definition at line 3071 of file STM8AF_STM8S.h.

5.1.2.1064 _TIM1_PSCRH

```
#define _TIM1_PSCRH _SFR(uint8_t, TIM1_AddressBase+0x10)
```

TIM1 clock prescaler register high byte.

Definition at line 2820 of file STM8AF_STM8S.h.

5.1.2.1065 _TIM1_PSCRH_RESET_VALUE

```
#define _TIM1_PSCRH_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 clock prescaler register high byte reset value.

Definition at line 2854 of file STM8AF_STM8S.h.

5.1.2.1066 _TIM1_PSCRL

```
#define _TIM1_PSCRL _SFR(uint8_t, TIM1_AddressBase+0x11)
```

TIM1 clock prescaler register low byte.

Definition at line 2821 of file STM8AF_STM8S.h.

5.1.2.1067 _TIM1_PSCRL_RESET_VALUE

```
#define _TIM1_PSCRL_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 clock prescaler register low byte reset value.

Definition at line 2855 of file STM8AF_STM8S.h.

5.1.2.1068 _TIM1_RCR

```
#define _TIM1_RCR _SFR(uint8_t, TIM1_AddressBase+0x14)
```

TIM1 Repetition counter.

Definition at line 2824 of file STM8AF_STM8S.h.

5.1.2.1069 _TIM1_RCR_RESET_VALUE

```
#define _TIM1_RCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 Repetition counter reset value.

Definition at line 2858 of file STM8AF_STM8S.h.

5.1.2.1070 _TIM1_SMCR

```
#define _TIM1_SMCR _SFR(uint8_t, TIM1_AddressBase+0x02)
```

TIM1 Slave mode control register.

Definition at line 2806 of file STM8AF_STM8S.h.

5.1.2.1071 _TIM1_SMCR_RESET_VALUE

```
#define _TIM1_SMCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 Slave mode control register reset value.

Definition at line 2840 of file STM8AF_STM8S.h.

5.1.2.1072 _TIM1_SMS

```
#define _TIM1_SMS ((uint8_t) (0x07 << 0))
```

TIM1 Clock/trigger/slave mode selection [2:0] (in _TIM1_SMCR)

Definition at line 2894 of file STM8AF_STM8S.h.

5.1.2.1073 _TIM1_SMS0

```
#define _TIM1_SMS0 ((uint8_t) (0x01 << 0))
```

TIM1 Clock/trigger/slave mode selection [0] (in _TIM1_SMCR)

Definition at line 2895 of file STM8AF_STM8S.h.

5.1.2.1074 _TIM1_SMS1

```
#define _TIM1_SMS1 ((uint8_t) (0x01 << 1))
```

TIM1 Clock/trigger/slave mode selection [1] (in _TIM1_SMCR)

Definition at line 2896 of file STM8AF_STM8S.h.

5.1.2.1075 _TIM1_SMS2

```
#define _TIM1_SMS2 ((uint8_t) (0x01 << 2))
```

TIM1 Clock/trigger/slave mode selection [2] (in _TIM1_SMCR)

Definition at line 2897 of file STM8AF_STM8S.h.

5.1.2.1076 _TIM1_SR1

```
#define _TIM1_SR1 _SFR(uint8_t, TIM1_AddressBase+0x05)
```

TIM1 status register 1.

Definition at line 2809 of file STM8AF_STM8S.h.

5.1.2.1077 _TIM1_SR1_RESET_VALUE

```
#define _TIM1_SR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 status register 1 reset value.

Definition at line 2843 of file STM8AF_STM8S.h.

5.1.2.1078 _TIM1_SR2

```
#define _TIM1_SR2 _SFR(uint8_t, TIM1_AddressBase+0x06)
```

TIM1 status register 2.

Definition at line 2810 of file STM8AF_STM8S.h.

5.1.2.1079 _TIM1_SR2_RESET_VALUE

```
#define _TIM1_SR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM1 status register 2 reset value.

Definition at line 2844 of file STM8AF_STM8S.h.

5.1.2.1080 _TIM1_TG

```
#define _TIM1_TG ((uint8_t) (0x01 << 6))
```

TIM1 Trigger generation [0] (in _TIM1_EGR)

Definition at line 2952 of file STM8AF_STM8S.h.

5.1.2.1081 _TIM1_TIE

```
#define _TIM1_TIE ((uint8_t) (0x01 << 6))
```

TIM1 Trigger interrupt enable [0] (in _TIM1_IER)

Definition at line 2924 of file STM8AF_STM8S.h.

5.1.2.1082 _TIM1_TIF

```
#define _TIM1_TIF ((uint8_t) (0x01 << 6))
```

TIM1 Trigger interrupt flag [0] (in _TIM1_SR1)

Definition at line 2934 of file STM8AF_STM8S.h.

5.1.2.1083 _TIM1_TS

```
#define _TIM1_TS ((uint8_t) (0x07 << 4))
```

TIM1 Trigger selection [2:0] (in _TIM1_SMCR)

Definition at line 2899 of file STM8AF_STM8S.h.

5.1.2.1084 _TIM1_TS0

```
#define _TIM1_TS0 ((uint8_t) (0x01 << 4))
```

TIM1 Trigger selection [0] (in _TIM1_SMCR)

Definition at line 2900 of file STM8AF_STM8S.h.

5.1.2.1085 _TIM1_TS1

```
#define _TIM1_TS1 ((uint8_t) (0x01 << 5))
```

TIM1 Trigger selection [1] (in _TIM1_SMCR)

Definition at line 2901 of file STM8AF_STM8S.h.

5.1.2.1086 _TIM1_TS2

```
#define _TIM1_TS2 ((uint8_t) (0x01 << 6))
```

TIM1 Trigger selection [2] (in _TIM1_SMCR)

Definition at line 2902 of file STM8AF_STM8S.h.

5.1.2.1087 _TIM1_UDIS

```
#define _TIM1_UDIS ((uint8_t) (0x01 << 1))
```

TIM1 Update disable [0] (in _TIM1_CR1)

Definition at line 2873 of file STM8AF_STM8S.h.

5.1.2.1088 _TIM1_UG

```
#define _TIM1_UG ((uint8_t) (0x01 << 0))
```

TIM1 Update generation [0] (in _TIM1_EGR)

Definition at line 2946 of file STM8AF_STM8S.h.

5.1.2.1089 _TIM1_UIE

```
#define _TIM1_UIE ((uint8_t) (0x01 << 0))
```

TIM1 Update interrupt enable [0] (in _TIM1_IER)

Definition at line 2918 of file STM8AF_STM8S.h.

5.1.2.1090 _TIM1_UIF

```
#define _TIM1_UIF ((uint8_t) (0x01 << 0))
```

TIM1 Update interrupt flag [0] (in _TIM1_SR1)

Definition at line 2928 of file STM8AF_STM8S.h.

5.1.2.1091 _TIM1_URS

```
#define _TIM1_URS ((uint8_t) (0x01 << 2))
```

TIM1 Update request source [0] (in _TIM1_CR1)

Definition at line 2874 of file STM8AF_STM8S.h.

5.1.2.1092 _TIM2

```
#define _TIM2 _SFR(TIM2_t, TIM2_AddressBase)
```

TIM2 struct/bit access.

Definition at line 3310 of file STM8AF_STM8S.h.

5.1.2.1093 _TIM2_ARPE

```
#define _TIM2_ARPE ((uint8_t) (0x01 << 7))
```

TIM2 Auto-reload preload enable [0] (in _TIM2_CR1)

Definition at line 3386 of file STM8AF_STM8S.h.

5.1.2.1094 _TIM2_ARRH

```
#define _TIM2_ARRH _SFR(uint8_t, TIM2_AddressBase+0x0D)
```

TIM2 auto-reload register high byte.

Definition at line 3347 of file STM8AF_STM8S.h.

5.1.2.1095 _TIM2_ARRH_RESET_VALUE

```
#define _TIM2_ARRH_RESET_VALUE ((uint8_t) 0xFF)
```

TIM2 auto-reload register high byte reset value.

Definition at line 3371 of file STM8AF_STM8S.h.

5.1.2.1096 _TIM2_ARRL

```
#define _TIM2_ARRL _SFR(uint8_t, TIM2_AddressBase+0x0E)
```

TIM2 auto-reload register low byte.

Definition at line 3348 of file STM8AF_STM8S.h.

5.1.2.1097 _TIM2_ARRL_RESET_VALUE

```
#define _TIM2_ARRL_RESET_VALUE ((uint8_t) 0xFF)
```

TIM2 auto-reload register low byte reset value.

Definition at line 3372 of file STM8AF_STM8S.h.

5.1.2.1098 _TIM2_CC1E

```
#define _TIM2_CC1E ((uint8_t) (0x01 << 0))
```

TIM2 Capture/compare 1 output enable [0] (in _TIM2_CCER1)

Definition at line 3486 of file STM8AF_STM8S.h.

5.1.2.1099 _TIM2_CC1G

```
#define _TIM2_CC1G ((uint8_t) (0x01 << 1))
```

TIM2 Capture/compare 1 generation [0] (in _TIM2_EGR)

Definition at line 3411 of file STM8AF_STM8S.h.

5.1.2.1100 _TIM2_CC1IE

```
#define _TIM2_CC1IE ((uint8_t) (0x01 << 1))
```

TIM2 Capture/compare 1 interrupt enable [0] (in _TIM2_IER)

Definition at line 3390 of file STM8AF_STM8S.h.

5.1.2.1101 _TIM2_CC1IF

```
#define _TIM2_CC1IF ((uint8_t) (0x01 << 1))
```

TIM2 Capture/compare 1 interrupt flag [0] (in _TIM2_SR1)

Definition at line 3397 of file STM8AF_STM8S.h.

5.1.2.1102 _TIM2_CC1OF

```
#define _TIM2_CC1OF ((uint8_t) (0x01 << 1))
```

TIM2 Capture/compare 1 overcapture flag [0] (in _TIM2_SR2)

Definition at line 3404 of file STM8AF_STM8S.h.

5.1.2.1103 _TIM2_CC1P

```
#define _TIM2_CC1P ((uint8_t) (0x01 << 1))
```

TIM2 Capture/compare 1 output polarity [0] (in _TIM2_CCER1)

Definition at line 3487 of file STM8AF_STM8S.h.

5.1.2.1104 _TIM2_CC1S

```
#define _TIM2_CC1S ((uint8_t) (0x03 << 0))
```

TIM2 Compare 1 selection [1:0] (in _TIM2_CCMR1)

Definition at line 3417 of file STM8AF_STM8S.h.

5.1.2.1105 _TIM2_CC1S0

```
#define _TIM2_CC1S0 ((uint8_t) (0x01 << 0))
```

TIM2 Compare 1 selection [0] (in _TIM2_CCMR1)

Definition at line 3418 of file STM8AF_STM8S.h.

5.1.2.1106 _TIM2_CC1S1

```
#define _TIM2_CC1S1 ((uint8_t) (0x01 << 1))
```

TIM2 Compare 1 selection [1] (in _TIM2_CCMR1)

Definition at line 3419 of file STM8AF_STM8S.h.

5.1.2.1107 _TIM2_CC2E

```
#define _TIM2_CC2E ((uint8_t) (0x01 << 4))
```

TIM2 Capture/compare 2 output enable [0] (in _TIM2_CCER1)

Definition at line 3489 of file STM8AF_STM8S.h.

5.1.2.1108 _TIM2_CC2G

```
#define _TIM2_CC2G ((uint8_t) (0x01 << 2))
```

TIM2 Capture/compare 2 generation [0] (in _TIM2_EGR)

Definition at line 3412 of file STM8AF_STM8S.h.

5.1.2.1109 _TIM2_CC2IE

```
#define _TIM2_CC2IE ((uint8_t) (0x01 << 2))
```

TIM2 Capture/compare 2 interrupt enable [0] (in _TIM2_IER)

Definition at line 3391 of file STM8AF_STM8S.h.

5.1.2.1110 _TIM2_CC2IF

```
#define _TIM2_CC2IF ((uint8_t) (0x01 << 2))
```

TIM2 Capture/compare 2 interrupt flag [0] (in _TIM2_SR1)

Definition at line 3398 of file STM8AF_STM8S.h.

5.1.2.1111 _TIM2_CC2OF

```
#define _TIM2_CC2OF ((uint8_t) (0x01 << 2))
```

TIM2 Capture/compare 2 overcapture flag [0] (in _TIM2_SR2)

Definition at line 3405 of file STM8AF_STM8S.h.

5.1.2.1112 _TIM2_CC2P

```
#define _TIM2_CC2P ((uint8_t) (0x01 << 5))
```

TIM2 Capture/compare 2 output polarity [0] (in _TIM2_CCER1)

Definition at line 3490 of file STM8AF_STM8S.h.

5.1.2.1113 _TIM2_CC2S

```
#define _TIM2_CC2S ((uint8_t) (0x03 << 0))
```

TIM2 Compare 2 selection [1:0] (in _TIM2_CCMR2)

Definition at line 3440 of file STM8AF_STM8S.h.

5.1.2.1114 _TIM2_CC2S0

```
#define _TIM2_CC2S0 ((uint8_t) (0x01 << 0))
```

TIM2 Compare 2 selection [0] (in _TIM2_CCMR2)

Definition at line 3441 of file STM8AF_STM8S.h.

5.1.2.1115 _TIM2_CC2S1

```
#define _TIM2_CC2S1 ((uint8_t) (0x01 << 1))
```

TIM2 Compare 2 selection [1] (in _TIM2_CCMR2)

Definition at line 3442 of file STM8AF_STM8S.h.

5.1.2.1116 _TIM2_CC3E

```
#define _TIM2_CC3E ((uint8_t) (0x01 << 0))
```

TIM2 Capture/compare 3 output enable [0] (in _TIM2_CCER2)

Definition at line 3494 of file STM8AF_STM8S.h.

5.1.2.1117 _TIM2_CC3G

```
#define _TIM2_CC3G ((uint8_t) (0x01 << 3))
```

TIM2 Capture/compare 3 generation [0] (in _TIM2_EGR)

Definition at line 3413 of file STM8AF_STM8S.h.

5.1.2.1118 _TIM2_CC3IE

```
#define _TIM2_CC3IE ((uint8_t) (0x01 << 3))
```

TIM2 Capture/compare 3 interrupt enable [0] (in _TIM2_IER)

Definition at line 3392 of file STM8AF_STM8S.h.

5.1.2.1119 _TIM2_CC3IF

```
#define _TIM2_CC3IF ((uint8_t) (0x01 << 3))
```

TIM2 Capture/compare 3 interrupt flag [0] (in _TIM2_SR1)

Definition at line 3399 of file STM8AF_STM8S.h.

5.1.2.1120 _TIM2_CC3OF

```
#define _TIM2_CC3OF ((uint8_t) (0x01 << 3))
```

TIM2 Capture/compare 3 overcapture flag [0] (in _TIM2_SR2)

Definition at line 3406 of file STM8AF_STM8S.h.

5.1.2.1121 _TIM2_CC3P

```
#define _TIM2_CC3P ((uint8_t) (0x01 << 1))
```

TIM2 Capture/compare 3 output polarity [0] (in _TIM2_CCER2)

Definition at line 3495 of file STM8AF_STM8S.h.

5.1.2.1122 _TIM2_CC3S

```
#define _TIM2_CC3S ((uint8_t) (0x03 << 0))
```

TIM2 Compare 3 selection [1:0] (in _TIM2_CCMR3)

Definition at line 3463 of file STM8AF_STM8S.h.

5.1.2.1123 _TIM2_CC3S0

```
#define _TIM2_CC3S0 ((uint8_t) (0x01 << 0))
```

TIM2 Compare 3 selection [0] (in _TIM2_CCMR3)

Definition at line 3464 of file STM8AF_STM8S.h.

5.1.2.1124 _TIM2_CC3S1

```
#define _TIM2_CC3S1 ((uint8_t) (0x01 << 1))
```

TIM2 Compare 3 selection [1] (in _TIM2_CCMR3)

Definition at line 3465 of file STM8AF_STM8S.h.

5.1.2.1125 _TIM2_CCER1

```
#define _TIM2_CCER1 _SFR(uint8_t, TIM2_AddressBase+0x08)
```

TIM2 Capture/compare enable register 1.

Definition at line 3342 of file STM8AF_STM8S.h.

5.1.2.1126 _TIM2_CCER1_RESET_VALUE

```
#define _TIM2_CCER1_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Capture/compare enable register 1 reset value.

Definition at line 3366 of file STM8AF_STM8S.h.

5.1.2.1127 _TIM2_CCER2

```
#define _TIM2_CCER2 _SFR(uint8_t, TIM2_AddressBase+0x09)
```

TIM2 Capture/compare enable register 2.

Definition at line 3343 of file STM8AF_STM8S.h.

5.1.2.1128 _TIM2_CCER2_RESET_VALUE

```
#define _TIM2_CCER2_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Capture/compare enable register 2 reset value.

Definition at line 3367 of file STM8AF_STM8S.h.

5.1.2.1129 _TIM2_CCMR1

```
#define _TIM2_CCMR1 _SFR(uint8_t, TIM2_AddressBase+0x05)
```

TIM2 Capture/compare mode register 1.

Definition at line 3339 of file STM8AF_STM8S.h.

5.1.2.1130 _TIM2_CCMR1_RESET_VALUE

```
#define _TIM2_CCMR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Capture/compare mode register 1 reset value.

Definition at line 3363 of file STM8AF_STM8S.h.

5.1.2.1131 _TIM2_CCMR2

```
#define _TIM2_CCMR2 _SFR(uint8_t, TIM2_AddressBase+0x06)
```

TIM2 Capture/compare mode register 2.

Definition at line 3340 of file STM8AF_STM8S.h.

5.1.2.1132 _TIM2_CCMR2_RESET_VALUE

```
#define _TIM2_CCMR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Capture/compare mode register 2 reset value.

Definition at line 3364 of file STM8AF_STM8S.h.

5.1.2.1133 _TIM2_CCMR3

```
#define _TIM2_CCMR3 _SFR(uint8_t, TIM2_AddressBase+0x07)
```

TIM2 Capture/compare mode register 3.

Definition at line 3341 of file STM8AF_STM8S.h.

5.1.2.1134 _TIM2_CCMR3_RESET_VALUE

```
#define _TIM2_CCMR3_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Capture/compare mode register 3 reset value.

Definition at line 3365 of file STM8AF_STM8S.h.

5.1.2.1135 _TIM2_CCR1H

```
#define _TIM2_CCR1H _SFR(uint8_t, TIM2_AddressBase+0x0F)
```

TIM2 16-bit capture/compare value 1 high byte.

Definition at line 3349 of file STM8AF_STM8S.h.

5.1.2.1136 _TIM2_CCR1H_RESET_VALUE

```
#define _TIM2_CCR1H_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 16-bit capture/compare value 1 high byte reset value.

Definition at line 3373 of file STM8AF_STM8S.h.

5.1.2.1137 _TIM2_CCR1L

```
#define _TIM2_CCR1L _SFR(uint8_t, TIM2_AddressBase+0x10)
```

TIM2 16-bit capture/compare value 1 low byte.

Definition at line 3350 of file STM8AF_STM8S.h.

5.1.2.1138 _TIM2_CCR1L_RESET_VALUE

```
#define _TIM2_CCR1L_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 16-bit capture/compare value 1 low byte reset value.

Definition at line 3374 of file STM8AF_STM8S.h.

5.1.2.1139 _TIM2_CCR2H

```
#define _TIM2_CCR2H _SFR(uint8_t, TIM2_AddressBase+0x11)
```

TIM2 16-bit capture/compare value 2 high byte.

Definition at line 3351 of file STM8AF_STM8S.h.

5.1.2.1140 _TIM2_CCR2H_RESET_VALUE

```
#define _TIM2_CCR2H_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 16-bit capture/compare value 2 high byte reset value.

Definition at line 3375 of file STM8AF_STM8S.h.

5.1.2.1141 _TIM2_CCR2L

```
#define _TIM2_CCR2L _SFR(uint8_t, TIM2_AddressBase+0x12)
```

TIM2 16-bit capture/compare value 2 low byte.

Definition at line 3352 of file STM8AF_STM8S.h.

5.1.2.1142 _TIM2_CCR2L_RESET_VALUE

```
#define _TIM2_CCR2L_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 16-bit capture/compare value 2 low byte reset value.

Definition at line 3376 of file STM8AF_STM8S.h.

5.1.2.1143 _TIM2_CCR3H

```
#define _TIM2_CCR3H _SFR(uint8_t, TIM2_AddressBase+0x13)
```

TIM2 16-bit capture/compare value 3 high byte.

Definition at line 3353 of file STM8AF_STM8S.h.

5.1.2.1144 _TIM2_CCR3H_RESET_VALUE

```
#define _TIM2_CCR3H_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 16-bit capture/compare value 3 high byte reset value.

Definition at line 3377 of file STM8AF_STM8S.h.

5.1.2.1145 _TIM2_CCR3L

```
#define _TIM2_CCR3L _SFR(uint8_t, TIM2_AddressBase+0x14)
```

TIM2 16-bit capture/compare value 3 low byte.

Definition at line 3354 of file STM8AF_STM8S.h.

5.1.2.1146 _TIM2_CCR3L_RESET_VALUE

```
#define _TIM2_CCR3L_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 16-bit capture/compare value 3 low byte reset value.

Definition at line 3378 of file STM8AF_STM8S.h.

5.1.2.1147 _TIM2_CEN

```
#define _TIM2_CEN ((uint8_t) (0x01 << 0))
```

TIM2 Counter enable [0] (in _TIM2_CR1)

Definition at line 3381 of file STM8AF_STM8S.h.

5.1.2.1148 _TIM2_CNTRH

```
#define _TIM2_CNTRH _SFR(uint8_t, TIM2_AddressBase+0x0A)
```

TIM2 counter register high byte.

Definition at line 3344 of file STM8AF_STM8S.h.

5.1.2.1149 _TIM2_CNTRH_RESET_VALUE

```
#define _TIM2_CNTRH_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 counter register high byte reset value.

Definition at line 3368 of file STM8AF_STM8S.h.

5.1.2.1150 _TIM2_CNTRL

```
#define _TIM2_CNTRL _SFR(uint8_t, TIM2_AddressBase+0x0B)
```

TIM2 counter register low byte.

Definition at line 3345 of file STM8AF_STM8S.h.

5.1.2.1151 _TIM2_CNTRL_RESET_VALUE

```
#define _TIM2_CNTRL_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 counter register low byte reset value.

Definition at line 3369 of file STM8AF_STM8S.h.

5.1.2.1152 _TIM2_CR1

```
#define _TIM2_CR1 _SFR(uint8_t, TIM2_AddressBase+0x00)
```

TIM2 control register 1.

Definition at line 3311 of file STM8AF_STM8S.h.

5.1.2.1153 _TIM2_CR1_RESET_VALUE

```
#define _TIM2_CR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 control register 1 reset value.

Definition at line 3358 of file STM8AF_STM8S.h.

5.1.2.1154 _TIM2_EGR

```
#define _TIM2_EGR _SFR(uint8_t, TIM2_AddressBase+0x04)
```

TIM2 Event generation register.

Definition at line 3338 of file STM8AF_STM8S.h.

5.1.2.1155 _TIM2_EGR_RESET_VALUE

```
#define _TIM2_EGR_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Event generation register reset value.

Definition at line 3362 of file STM8AF_STM8S.h.

5.1.2.1156 _TIM2_IC1F

```
#define _TIM2_IC1F ((uint8_t) (0x0F << 4))
```

TIM2 Output compare 1 mode [3:0] (in _TIM2_CCMR1)

Definition at line 3433 of file STM8AF_STM8S.h.

5.1.2.1157 _TIM2_IC1F0

```
#define _TIM2_IC1F0 ((uint8_t) (0x01 << 4))
```

TIM2 Output compare 1 mode [0] (in _TIM2_CCMR1)

Definition at line 3434 of file STM8AF_STM8S.h.

5.1.2.1158 _TIM2_IC1F1

```
#define _TIM2_IC1F1 ((uint8_t) (0x01 << 5))
```

TIM2 Output compare 1 mode [1] (in _TIM2_CCMR1)

Definition at line 3435 of file STM8AF_STM8S.h.

5.1.2.1159 _TIM2_IC1F2

```
#define _TIM2_IC1F2 ((uint8_t) (0x01 << 6))
```

TIM2 Output compare 1 mode [2] (in _TIM2_CCMR1)

Definition at line 3436 of file STM8AF_STM8S.h.

5.1.2.1160 _TIM2_IC1F3

```
#define _TIM2_IC1F3 ((uint8_t) (0x01 << 7))
```

TIM2 Output compare 1 mode [3] (in _TIM2_CCMR1)

Definition at line 3437 of file STM8AF_STM8S.h.

5.1.2.1161 _TIM2_IC1PSC

```
#define _TIM2_IC1PSC ((uint8_t) (0x03 << 2))
```

TIM2 Input capture 1 prescaler [1:0] (in _TIM2_CCMR1)

Definition at line 3430 of file STM8AF_STM8S.h.

5.1.2.1162 _TIM2_IC1PSC0

```
#define _TIM2_IC1PSC0 ((uint8_t) (0x01 << 2))
```

TIM2 Input capture 1 prescaler [0] (in _TIM2_CCMR1)

Definition at line 3431 of file STM8AF_STM8S.h.

5.1.2.1163 _TIM2_IC1PSC1

```
#define _TIM2_IC1PSC1 ((uint8_t) (0x01 << 3))
```

TIM2 Input capture 1 prescaler [1] (in _TIM2_CCMR1)

Definition at line 3432 of file STM8AF_STM8S.h.

5.1.2.1164 _TIM2_IC2F

```
#define _TIM2_IC2F ((uint8_t) (0x0F << 4))
```

TIM2 Output compare 2 mode [3:0] (in _TIM2_CCMR2)

Definition at line 3456 of file STM8AF_STM8S.h.

5.1.2.1165 _TIM2_IC2F0

```
#define _TIM2_IC2F0 ((uint8_t) (0x01 << 4))
```

TIM2 Output compare 2 mode [0] (in _TIM2_CCMR2)

Definition at line 3457 of file STM8AF_STM8S.h.

5.1.2.1166 _TIM2_IC2F1

```
#define _TIM2_IC2F1 ((uint8_t) (0x01 << 5))
```

TIM2 Output compare 2 mode [1] (in _TIM2_CCMR2)

Definition at line 3458 of file STM8AF_STM8S.h.

5.1.2.1167 _TIM2_IC2F2

```
#define _TIM2_IC2F2 ((uint8_t) (0x01 << 6))
```

TIM2 Output compare 2 mode [2] (in _TIM2_CCMR2)

Definition at line 3459 of file STM8AF_STM8S.h.

5.1.2.1168 _TIM2_IC2F3

```
#define _TIM2_IC2F3 ((uint8_t) (0x01 << 7))
```

TIM2 Output compare 2 mode [3] (in _TIM2_CCMR2)

Definition at line 3460 of file STM8AF_STM8S.h.

5.1.2.1169 _TIM2_IC2PSC

```
#define _TIM2_IC2PSC ((uint8_t) (0x03 << 2))
```

TIM2 Input capture 2 prescaler [1:0] (in _TIM2_CCMR2)

Definition at line 3453 of file STM8AF_STM8S.h.

5.1.2.1170 _TIM2_IC2PSC0

```
#define _TIM2_IC2PSC0 ((uint8_t) (0x01 << 2))
```

TIM2 Input capture 2 prescaler [0] (in _TIM2_CCMR2)

Definition at line 3454 of file STM8AF_STM8S.h.

5.1.2.1171 _TIM2_IC2PSC1

```
#define _TIM2_IC2PSC1 ((uint8_t) (0x01 << 3))
```

TIM2 Input capture 2 prescaler [1] (in _TIM2_CCMR2)

Definition at line 3455 of file STM8AF_STM8S.h.

5.1.2.1172 _TIM2_IC3F

```
#define _TIM2_IC3F ((uint8_t) (0x0F << 4))
```

TIM2 Output compare 3 mode [3:0] (in _TIM2_CCMR3)

Definition at line 3479 of file STM8AF_STM8S.h.

5.1.2.1173 _TIM2_IC3F0

```
#define _TIM2_IC3F0 ((uint8_t) (0x01 << 4))
```

TIM2 Output compare 3 mode [0] (in _TIM2_CCMR3)

Definition at line 3480 of file STM8AF_STM8S.h.

5.1.2.1174 _TIM2_IC3F1

```
#define _TIM2_IC3F1 ((uint8_t) (0x01 << 5))
```

TIM2 Output compare 3 mode [1] (in _TIM2_CCMR3)

Definition at line 3481 of file STM8AF_STM8S.h.

5.1.2.1175 _TIM2_IC3F2

```
#define _TIM2_IC3F2 ((uint8_t) (0x01 << 6))
```

TIM2 Output compare 3 mode [2] (in _TIM2_CCMR3)

Definition at line 3482 of file STM8AF_STM8S.h.

5.1.2.1176 _TIM2_IC3F3

```
#define _TIM2_IC3F3 ((uint8_t) (0x01 << 7))
```

TIM2 Output compare 3 mode [3] (in _TIM2_CCMR3)

Definition at line 3483 of file STM8AF_STM8S.h.

5.1.2.1177 _TIM2_IC3PSC

```
#define _TIM2_IC3PSC ((uint8_t) (0x03 << 2))
```

TIM2 Input capture 3 prescaler [1:0] (in _TIM2_CCMR3)

Definition at line 3476 of file STM8AF_STM8S.h.

5.1.2.1178 _TIM2_IC3PSC0

```
#define _TIM2_IC3PSC0 ((uint8_t) (0x01 << 2))
```

TIM2 Input capture 3 prescaler [0] (in _TIM2_CCMR3)

Definition at line 3477 of file STM8AF_STM8S.h.

5.1.2.1179 _TIM2_IC3PSC1

```
#define _TIM2_IC3PSC1 ((uint8_t) (0x01 << 3))
```

TIM2 Input capture 3 prescaler [1] (in _TIM2_CCMR3)

Definition at line 3478 of file STM8AF_STM8S.h.

5.1.2.1180 _TIM2_IER

```
#define _TIM2_IER _SFR(uint8_t, TIM2_AddressBase+0x01)
```

TIM2 interrupt enable register.

Definition at line 3335 of file STM8AF_STM8S.h.

5.1.2.1181 _TIM2_IER_RESET_VALUE

```
#define _TIM2_IER_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 interrupt enable register reset value.

Definition at line 3359 of file STM8AF_STM8S.h.

5.1.2.1182 _TIM2_OC1M

```
#define _TIM2_OC1M ((uint8_t) (0x07 << 4))
```

TIM2 Output compare 1 mode [2:0] (in _TIM2_CCMR1)

Definition at line 3422 of file STM8AF_STM8S.h.

5.1.2.1183 _TIM2_OC1M0

```
#define _TIM2_OC1M0 ((uint8_t) (0x01 << 4))
```

TIM2 Output compare 1 mode [0] (in _TIM2_CCMR1)

Definition at line 3423 of file STM8AF_STM8S.h.

5.1.2.1184 _TIM2_OC1M1

```
#define _TIM2_OC1M1 ((uint8_t) (0x01 << 5))
```

TIM2 Output compare 1 mode [1] (in _TIM2_CCMR1)

Definition at line 3424 of file STM8AF_STM8S.h.

5.1.2.1185 _TIM2_OC1M2

```
#define _TIM2_OC1M2 ((uint8_t) (0x01 << 6))
```

TIM2 Output compare 1 mode [2] (in _TIM2_CCMR1)

Definition at line 3425 of file STM8AF_STM8S.h.

5.1.2.1186 _TIM2_OC1PE

```
#define _TIM2_OC1PE ((uint8_t) (0x01 << 3))
```

TIM2 Output compare 1 preload enable [0] (in _TIM2_CCMR1)

Definition at line 3421 of file STM8AF_STM8S.h.

5.1.2.1187 _TIM2_OC2M

```
#define _TIM2_OC2M ((uint8_t) (0x07 << 4))
```

TIM2 Output compare 2 mode [2:0] (in _TIM2_CCMR2)

Definition at line 3445 of file STM8AF_STM8S.h.

5.1.2.1188 _TIM2_OC2M0

```
#define _TIM2_OC2M0 ((uint8_t) (0x01 << 4))
```

TIM2 Output compare 2 mode [0] (in _TIM2_CCMR2)

Definition at line 3446 of file STM8AF_STM8S.h.

5.1.2.1189 _TIM2_OC2M1

```
#define _TIM2_OC2M1 ((uint8_t) (0x01 << 5))
```

TIM2 Output compare 2 mode [1] (in _TIM2_CCMR2)

Definition at line 3447 of file STM8AF_STM8S.h.

5.1.2.1190 _TIM2_OC2M2

```
#define _TIM2_OC2M2 ((uint8_t) (0x01 << 6))
```

TIM2 Output compare 2 mode [2] (in _TIM2_CCMR2)

Definition at line 3448 of file STM8AF_STM8S.h.

5.1.2.1191 _TIM2_OC2PE

```
#define _TIM2_OC2PE ((uint8_t) (0x01 << 3))
```

TIM2 Output compare 2 preload enable [0] (in _TIM2_CCMR2)

Definition at line 3444 of file STM8AF_STM8S.h.

5.1.2.1192 _TIM2_OC3M

```
#define _TIM2_OC3M ((uint8_t) (0x07 << 4))
```

TIM2 Output compare 3 mode [2:0] (in _TIM2_CCMR3)

Definition at line 3468 of file STM8AF_STM8S.h.

5.1.2.1193 _TIM2_OC3M0

```
#define _TIM2_OC3M0 ((uint8_t) (0x01 << 4))
```

TIM2 Output compare 3 mode [0] (in _TIM2_CCMR3)

Definition at line 3469 of file STM8AF_STM8S.h.

5.1.2.1194 _TIM2_OC3M1

```
#define _TIM2_OC3M1 ((uint8_t) (0x01 << 5))
```

TIM2 Output compare 3 mode [1] (in _TIM2_CCMR3)

Definition at line 3470 of file STM8AF_STM8S.h.

5.1.2.1195 _TIM2_OC3M2

```
#define _TIM2_OC3M2 ((uint8_t) (0x01 << 6))
```

TIM2 Output compare 3 mode [2] (in _TIM2_CCMR3)

Definition at line 3471 of file STM8AF_STM8S.h.

5.1.2.1196 _TIM2_OC3PE

```
#define _TIM2_OC3PE ((uint8_t) (0x01 << 3))
```

TIM2 Output compare 3 preload enable [0] (in _TIM2_CCMR3)

Definition at line 3467 of file STM8AF_STM8S.h.

5.1.2.1197 _TIM2_OPM

```
#define _TIM2_OPM ((uint8_t) (0x01 << 3))
```

TIM2 One-pulse mode [0] (in _TIM2_CR1)

Definition at line 3384 of file STM8AF_STM8S.h.

5.1.2.1198 _TIM2_PSC

```
#define _TIM2_PSC ((uint8_t) (0x0F << 0))
```

TIM2 prescaler [3:0] (in _TIM2_PSCR)

Definition at line 3499 of file STM8AF_STM8S.h.

5.1.2.1199 _TIM2_PSC0

```
#define _TIM2_PSC0 ((uint8_t) (0x01 << 0))
```

TIM2 prescaler [0] (in _TIM2_PSCR)

Definition at line 3500 of file STM8AF_STM8S.h.

5.1.2.1200 _TIM2_PSC1

```
#define _TIM2_PSC1 ((uint8_t) (0x01 << 1))
```

TIM2 prescaler [1] (in _TIM2_PSCR)

Definition at line 3501 of file STM8AF_STM8S.h.

5.1.2.1201 _TIM2_PSC2

```
#define _TIM2_PSC2 ((uint8_t) (0x01 << 2))
```

TIM2 prescaler [2] (in _TIM2_PSCR)

Definition at line 3502 of file STM8AF_STM8S.h.

5.1.2.1202 _TIM2_PSC3

```
#define _TIM2_PSC3 ((uint8_t) (0x01 << 3))
```

TIM2 prescaler [3] (in _TIM2_PSCR)

Definition at line 3503 of file STM8AF_STM8S.h.

5.1.2.1203 _TIM2_PSCR

```
#define _TIM2_PSCR _SFR(uint8_t, TIM2_AddressBase+0x0C)
```

TIM2 clock prescaler register.

Definition at line 3346 of file STM8AF_STM8S.h.

5.1.2.1204 _TIM2_PSCR_RESET_VALUE

```
#define _TIM2_PSCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 clock prescaler register reset value.

Definition at line 3370 of file STM8AF_STM8S.h.

5.1.2.1205 _TIM2_SR1

```
#define _TIM2_SR1 _SFR(uint8_t, TIM2_AddressBase+0x02)
```

TIM2 status register 1.

Definition at line 3336 of file STM8AF_STM8S.h.

5.1.2.1206 _TIM2_SR1_RESET_VALUE

```
#define _TIM2_SR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 status register 1 reset value.

Definition at line 3360 of file STM8AF_STM8S.h.

5.1.2.1207 _TIM2_SR2

```
#define _TIM2_SR2 _SFR(uint8_t, TIM2_AddressBase+0x03)
```

TIM2 status register 2.

Definition at line 3337 of file STM8AF_STM8S.h.

5.1.2.1208 _TIM2_SR2_RESET_VALUE

```
#define _TIM2_SR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 status register 2 reset value.

Definition at line 3361 of file STM8AF_STM8S.h.

5.1.2.1209 _TIM2_UDIS

```
#define _TIM2_UDIS ((uint8_t) (0x01 << 1))
```

TIM2 Update disable [0] (in _TIM2_CR1)

Definition at line 3382 of file STM8AF_STM8S.h.

5.1.2.1210 _TIM2_UG

```
#define _TIM2_UG ((uint8_t) (0x01 << 0))
```

TIM2 Update generation [0] (in _TIM2_EGR)

Definition at line 3410 of file STM8AF_STM8S.h.

5.1.2.1211 _TIM2_UIE

```
#define _TIM2_UIE ((uint8_t) (0x01 << 0))
```

TIM2 Update interrupt enable [0] (in _TIM2_IER)

Definition at line 3389 of file STM8AF_STM8S.h.

5.1.2.1212 _TIM2_UIF

```
#define _TIM2_UIF ((uint8_t) (0x01 << 0))
```

TIM2 Update interrupt flag [0] (in _TIM2_SR1)

Definition at line 3396 of file STM8AF_STM8S.h.

5.1.2.1213 _TIM2_URS

```
#define _TIM2_URS ((uint8_t) (0x01 << 2))
```

TIM2 Update request source [0] (in _TIM2_CR1)

Definition at line 3383 of file STM8AF_STM8S.h.

5.1.2.1214 _TIM3

```
#define _TIM3 _SFR(TIM3_t, TIM3_AddressBase)
```

TIM3 struct/bit access.

Definition at line 3677 of file STM8AF_STM8S.h.

5.1.2.1215 _TIM3_ARPE

```
#define _TIM3_ARPE ((uint8_t) (0x01 << 7))
```

TIM3 Auto-reload preload enable [0] (in _TIM3_CR1)

Definition at line 3721 of file STM8AF_STM8S.h.

5.1.2.1216 _TIM3_ARRH

```
#define _TIM3_ARRH _SFR(uint8_t, TIM3_AddressBase+0x0D)
```

TIM3 auto-reload register high byte.

Definition at line 3689 of file STM8AF_STM8S.h.

5.1.2.1217 _TIM3_ARRH_RESET_VALUE

```
#define _TIM3_ARRH_RESET_VALUE ((uint8_t) 0xFF)
```

TIM3 auto-reload register high byte reset value.

Definition at line 3708 of file STM8AF_STM8S.h.

5.1.2.1218 _TIM3_ARRL

```
#define _TIM3_ARRL _SFR(uint8_t, TIM3_AddressBase+0x0E)
```

TIM3 auto-reload register low byte.

Definition at line 3690 of file STM8AF_STM8S.h.

5.1.2.1219 _TIM3_ARRL_RESET_VALUE

```
#define _TIM3_ARRL_RESET_VALUE ((uint8_t) 0xFF)
```

TIM3 auto-reload register low byte reset value.

Definition at line 3709 of file STM8AF_STM8S.h.

5.1.2.1220 _TIM3_CC1E

```
#define _TIM3_CC1E ((uint8_t) (0x01 << 0))
```

TIM3 Capture/compare 1 output enable [0] (in _TIM3_CCER1)

Definition at line 3794 of file STM8AF_STM8S.h.

5.1.2.1221 _TIM3_CC1G

```
#define _TIM3_CC1G ((uint8_t) (0x01 << 1))
```

TIM3 Capture/compare 1 generation [0] (in _TIM3_EGR)

Definition at line 3743 of file STM8AF_STM8S.h.

5.1.2.1222 _TIM3_CC1IE

```
#define _TIM3_CC1IE ((uint8_t) (0x01 << 1))
```

TIM3 Capture/compare 1 interrupt enable [0] (in _TIM3_IER)

Definition at line 3725 of file STM8AF_STM8S.h.

5.1.2.1223 _TIM3_CC1IF

```
#define _TIM3_CC1IF ((uint8_t) (0x01 << 1))
```

TIM3 Capture/compare 1 interrupt flag [0] (in _TIM3_SR1)

Definition at line 3731 of file STM8AF_STM8S.h.

5.1.2.1224 _TIM3_CC1OF

```
#define _TIM3_CC1OF ((uint8_t) (0x01 << 1))
```

TIM3 Capture/compare 1 overcapture flag [0] (in _TIM3_SR2)

Definition at line 3737 of file STM8AF_STM8S.h.

5.1.2.1225 _TIM3_CC1P

```
#define _TIM3_CC1P ((uint8_t) (0x01 << 1))
```

TIM3 Capture/compare 1 output polarity [0] (in _TIM3_CCER1)

Definition at line 3795 of file STM8AF_STM8S.h.

5.1.2.1226 _TIM3_CC1S

```
#define _TIM3_CC1S ((uint8_t) (0x03 << 0))
```

TIM3 Compare 1 selection [1:0] (in _TIM3_CCMR1)

Definition at line 3748 of file STM8AF_STM8S.h.

5.1.2.1227 _TIM3_CC1S0

```
#define _TIM3_CC1S0 ((uint8_t) (0x01 << 0))
```

TIM3 Compare 1 selection [0] (in _TIM3_CCMR1)

Definition at line 3749 of file STM8AF_STM8S.h.

5.1.2.1228 _TIM3_CC1S1

```
#define _TIM3_CC1S1 ((uint8_t) (0x01 << 1))
```

TIM3 Compare 1 selection [1] (in _TIM3_CCMR1)

Definition at line 3750 of file STM8AF_STM8S.h.

5.1.2.1229 _TIM3_CC2E

```
#define _TIM3_CC2E ((uint8_t) (0x01 << 4))
```

TIM3 Capture/compare 2 output enable [0] (in _TIM3_CCER1)

Definition at line 3797 of file STM8AF_STM8S.h.

5.1.2.1230 _TIM3_CC2G

```
#define _TIM3_CC2G ((uint8_t) (0x01 << 2))
```

TIM3 Capture/compare 2 generation [0] (in _TIM3_EGR)

Definition at line 3744 of file STM8AF_STM8S.h.

5.1.2.1231 _TIM3_CC2IE

```
#define _TIM3_CC2IE ((uint8_t) (0x01 << 2))
```

TIM3 Capture/compare 2 interrupt enable [0] (in _TIM3_IER)

Definition at line 3726 of file STM8AF_STM8S.h.

5.1.2.1232 _TIM3_CC2IF

```
#define _TIM3_CC2IF ((uint8_t) (0x01 << 2))
```

TIM3 Capture/compare 2 interrupt flag [0] (in _TIM3_SR1)

Definition at line 3732 of file STM8AF_STM8S.h.

5.1.2.1233 _TIM3_CC2OF

```
#define _TIM3_CC2OF ((uint8_t) (0x01 << 2))
```

TIM3 Capture/compare 2 overcapture flag [0] (in _TIM3_SR2)

Definition at line 3738 of file STM8AF_STM8S.h.

5.1.2.1234 _TIM3_CC2P

```
#define _TIM3_CC2P ((uint8_t) (0x01 << 5))
```

TIM3 Capture/compare 2 output polarity [0] (in _TIM3_CCER1)

Definition at line 3798 of file STM8AF_STM8S.h.

5.1.2.1235 _TIM3_CC2S

```
#define _TIM3_CC2S ((uint8_t) (0x03 << 0))
```

TIM3 Compare 2 selection [1:0] (in _TIM3_CCMR2)

Definition at line 3771 of file STM8AF_STM8S.h.

5.1.2.1236 _TIM3_CC2S0

```
#define _TIM3_CC2S0 ((uint8_t) (0x01 << 0))
```

TIM3 Compare 2 selection [0] (in _TIM3_CCMR2)

Definition at line 3772 of file STM8AF_STM8S.h.

5.1.2.1237 _TIM3_CC2S1

```
#define _TIM3_CC2S1 ((uint8_t) (0x01 << 1))
```

TIM3 Compare 2 selection [1] (in _TIM3_CCMR2)

Definition at line 3773 of file STM8AF_STM8S.h.

5.1.2.1238 _TIM3_CCER1

```
#define _TIM3_CCER1 _SFR(uint8_t, TIM3_AddressBase+0x08)
```

TIM3 Capture/compare enable register 1.

Definition at line 3685 of file STM8AF_STM8S.h.

5.1.2.1239 _TIM3_CCER1_RESET_VALUE

```
#define _TIM3_CCER1_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 Capture/compare enable register 1 reset value.

Definition at line 3704 of file STM8AF_STM8S.h.

5.1.2.1240 _TIM3_CCMR1

```
#define _TIM3_CCMR1 _SFR(uint8_t, TIM3_AddressBase+0x05)
```

TIM3 Capture/compare mode register 1.

Definition at line 3683 of file STM8AF_STM8S.h.

5.1.2.1241 _TIM3_CCMR1_RESET_VALUE

```
#define _TIM3_CCMR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 Capture/compare mode register 1 reset value.

Definition at line 3702 of file STM8AF_STM8S.h.

5.1.2.1242 _TIM3_CCMR2

```
#define _TIM3_CCMR2 _SFR(uint8_t, TIM3_AddressBase+0x06)
```

TIM3 Capture/compare mode register 2.

Definition at line 3684 of file STM8AF_STM8S.h.

5.1.2.1243 _TIM3_CCMR2_RESET_VALUE

```
#define _TIM3_CCMR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 Capture/compare mode register 2 reset value.

Definition at line 3703 of file STM8AF_STM8S.h.

5.1.2.1244 _TIM3_CCR1H

```
#define _TIM3_CCR1H _SFR(uint8_t, TIM3_AddressBase+0x0F)
```

TIM3 16-bit capture/compare value 1 high byte.

Definition at line 3691 of file STM8AF_STM8S.h.

5.1.2.1245 _TIM3_CCR1H_RESET_VALUE

```
#define _TIM3_CCR1H_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 16-bit capture/compare value 1 high byte reset value.

Definition at line 3710 of file STM8AF_STM8S.h.

5.1.2.1246 _TIM3_CCR1L

```
#define _TIM3_CCR1L _SFR(uint8_t, TIM3_AddressBase+0x10)
```

TIM3 16-bit capture/compare value 1 low byte.

Definition at line 3692 of file STM8AF_STM8S.h.

5.1.2.1247 _TIM3_CCR1L_RESET_VALUE

```
#define _TIM3_CCR1L_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 16-bit capture/compare value 1 low byte reset value.

Definition at line 3711 of file STM8AF_STM8S.h.

5.1.2.1248 _TIM3_CCR2H

```
#define _TIM3_CCR2H _SFR(uint8_t, TIM3_AddressBase+0x11)
```

TIM3 16-bit capture/compare value 2 high byte.

Definition at line 3693 of file STM8AF_STM8S.h.

5.1.2.1249 _TIM3_CCR2H_RESET_VALUE

```
#define _TIM3_CCR2H_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 16-bit capture/compare value 2 high byte reset value.

Definition at line 3712 of file STM8AF_STM8S.h.

5.1.2.1250 _TIM3_CCR2L

```
#define _TIM3_CCR2L _SFR(uint8_t, TIM3_AddressBase+0x12)
```

TIM3 16-bit capture/compare value 2 low byte.

Definition at line 3694 of file STM8AF_STM8S.h.

5.1.2.1251 _TIM3_CCR2L_RESET_VALUE

```
#define _TIM3_CCR2L_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 16-bit capture/compare value 2 low byte reset value.

Definition at line 3713 of file STM8AF_STM8S.h.

5.1.2.1252 _TIM3_CEN

```
#define _TIM3_CEN ((uint8_t) (0x01 << 0))
```

TIM3 Counter enable [0] (in _TIM3_CR1)

Definition at line 3716 of file STM8AF_STM8S.h.

5.1.2.1253 _TIM3_CNTRH

```
#define _TIM3_CNTRH _SFR(uint8_t, TIM3_AddressBase+0x0A)
```

TIM3 counter register high byte.

Definition at line 3686 of file STM8AF_STM8S.h.

5.1.2.1254 _TIM3_CNTRH_RESET_VALUE

```
#define _TIM3_CNTRH_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 counter register high byte reset value.

Definition at line 3705 of file STM8AF_STM8S.h.

5.1.2.1255 _TIM3_CNTRL

```
#define _TIM3_CNTRL _SFR(uint8_t, TIM3_AddressBase+0x0B)
```

TIM3 counter register low byte.

Definition at line 3687 of file STM8AF_STM8S.h.

5.1.2.1256 _TIM3_CNTRL_RESET_VALUE

```
#define _TIM3_CNTRL_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 counter register low byte reset value.

Definition at line 3706 of file STM8AF_STM8S.h.

5.1.2.1257 _TIM3_CR1

```
#define _TIM3_CR1 _SFR(uint8_t, TIM3_AddressBase+0x00)
```

TIM3 control register 1.

Definition at line 3678 of file STM8AF_STM8S.h.

5.1.2.1258 _TIM3_CR1_RESET_VALUE

```
#define _TIM3_CR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 control register 1 reset value.

Definition at line 3697 of file STM8AF_STM8S.h.

5.1.2.1259 _TIM3_EGR

```
#define _TIM3_EGR _SFR(uint8_t, TIM3_AddressBase+0x04)
```

TIM3 Event generation register.

Definition at line 3682 of file STM8AF_STM8S.h.

5.1.2.1260 _TIM3_EGR_RESET_VALUE

```
#define _TIM3_EGR_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 Event generation register reset value.

Definition at line 3701 of file STM8AF_STM8S.h.

5.1.2.1261 _TIM3_IC1F

```
#define _TIM3_IC1F ((uint8_t) (0x0F << 4))
```

TIM3 Output compare 1 mode [3:0] (in _TIM3_CCMR1)

Definition at line 3764 of file STM8AF_STM8S.h.

5.1.2.1262 _TIM3_IC1F0

```
#define _TIM3_IC1F0 ((uint8_t) (0x01 << 4))
```

TIM3 Output compare 1 mode [0] (in _TIM3_CCMR1)

Definition at line 3765 of file STM8AF_STM8S.h.

5.1.2.1263 _TIM3_IC1F1

```
#define _TIM3_IC1F1 ((uint8_t) (0x01 << 5))
```

TIM3 Output compare 1 mode [1] (in _TIM3_CCMR1)

Definition at line 3766 of file STM8AF_STM8S.h.

5.1.2.1264 _TIM3_IC1F2

```
#define _TIM3_IC1F2 ((uint8_t) (0x01 << 6))
```

TIM3 Output compare 1 mode [2] (in _TIM3_CCMR1)

Definition at line 3767 of file STM8AF_STM8S.h.

5.1.2.1265 _TIM3_IC1F3

```
#define _TIM3_IC1F3 ((uint8_t) (0x01 << 7))
```

TIM3 Output compare 1 mode [3] (in _TIM3_CCMR1)

Definition at line 3768 of file STM8AF_STM8S.h.

5.1.2.1266 _TIM3_IC1PSC

```
#define _TIM3_IC1PSC ((uint8_t) (0x03 << 2))
```

TIM3 Input capture 1 prescaler [1:0] (in _TIM3_CCMR1)

Definition at line 3761 of file STM8AF_STM8S.h.

5.1.2.1267 _TIM3_IC1PSC0

```
#define _TIM3_IC1PSC0 ((uint8_t) (0x01 << 2))
```

TIM3 Input capture 1 prescaler [0] (in _TIM3_CCMR1)

Definition at line 3762 of file STM8AF_STM8S.h.

5.1.2.1268 _TIM3_IC1PSC1

```
#define _TIM3_IC1PSC1 ((uint8_t) (0x01 << 3))
```

TIM3 Input capture 1 prescaler [1] (in _TIM3_CCMR1)

Definition at line 3763 of file STM8AF_STM8S.h.

5.1.2.1269 _TIM3_IC2F

```
#define _TIM3_IC2F ((uint8_t) (0x0F << 4))
```

TIM3 Output compare 2 mode [3:0] (in _TIM3_CCMR2)

Definition at line 3787 of file STM8AF_STM8S.h.

5.1.2.1270 _TIM3_IC2F0

```
#define _TIM3_IC2F0 ((uint8_t) (0x01 << 4))
```

TIM3 Output compare 2 mode [0] (in _TIM3_CCMR2)

Definition at line 3788 of file STM8AF_STM8S.h.

5.1.2.1271 _TIM3_IC2F1

```
#define _TIM3_IC2F1 ((uint8_t) (0x01 << 5))
```

TIM3 Output compare 2 mode [1] (in _TIM3_CCMR2)

Definition at line 3789 of file STM8AF_STM8S.h.

5.1.2.1272 _TIM3_IC2F2

```
#define _TIM3_IC2F2 ((uint8_t) (0x01 << 6))
```

TIM3 Output compare 2 mode [2] (in _TIM3_CCMR2)

Definition at line 3790 of file STM8AF_STM8S.h.

5.1.2.1273 _TIM3_IC2F3

```
#define _TIM3_IC2F3 ((uint8_t) (0x01 << 7))
```

TIM3 Output compare 2 mode [3] (in _TIM3_CCMR2)

Definition at line 3791 of file STM8AF_STM8S.h.

5.1.2.1274 _TIM3_IC2PSC

```
#define _TIM3_IC2PSC ((uint8_t) (0x03 << 2))
```

TIM3 Input capture 2 prescaler [1:0] (in _TIM3_CCMR2)

Definition at line 3784 of file STM8AF_STM8S.h.

5.1.2.1275 _TIM3_IC2PSC0

```
#define _TIM3_IC2PSC0 ((uint8_t) (0x01 << 2))
```

TIM3 Input capture 2 prescaler [0] (in _TIM3_CCMR2)

Definition at line 3785 of file STM8AF_STM8S.h.

5.1.2.1276 _TIM3_IC2PSC1

```
#define _TIM3_IC2PSC1 ((uint8_t) (0x01 << 3))
```

TIM3 Input capture 2 prescaler [1] (in _TIM3_CCMR2)

Definition at line 3786 of file STM8AF_STM8S.h.

5.1.2.1277 _TIM3_IER

```
#define _TIM3_IER _SFR(uint8_t, TIM3_AddressBase+0x01)
```

TIM3 interrupt enable register.

Definition at line 3679 of file STM8AF_STM8S.h.

5.1.2.1278 _TIM3_IER_RESET_VALUE

```
#define _TIM3_IER_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 interrupt enable register reset value.

Definition at line 3698 of file STM8AF_STM8S.h.

5.1.2.1279 _TIM3_OC1M

```
#define _TIM3_OC1M ((uint8_t) (0x07 << 4))
```

TIM3 Output compare 1 mode [2:0] (in _TIM3_CCMR1)

Definition at line 3753 of file STM8AF_STM8S.h.

5.1.2.1280 _TIM3_OC1M0

```
#define _TIM3_OC1M0 ((uint8_t) (0x01 << 4))
```

TIM3 Output compare 1 mode [0] (in _TIM3_CCMR1)

Definition at line 3754 of file STM8AF_STM8S.h.

5.1.2.1281 _TIM3_OC1M1

```
#define _TIM3_OC1M1 ((uint8_t) (0x01 << 5))
```

TIM3 Output compare 1 mode [1] (in _TIM3_CCMR1)

Definition at line 3755 of file STM8AF_STM8S.h.

5.1.2.1282 _TIM3_OC1M2

```
#define _TIM3_OC1M2 ((uint8_t) (0x01 << 6))
```

TIM3 Output compare 1 mode [2] (in _TIM3_CCMR1)

Definition at line 3756 of file STM8AF_STM8S.h.

5.1.2.1283 _TIM3_OC1PE

```
#define _TIM3_OC1PE ((uint8_t) (0x01 << 3))
```

TIM3 Output compare 1 preload enable [0] (in _TIM3_CCMR1)

Definition at line 3752 of file STM8AF_STM8S.h.

5.1.2.1284 _TIM3_OC2M

```
#define _TIM3_OC2M ((uint8_t) (0x07 << 4))
```

TIM3 Output compare 2 mode [2:0] (in _TIM3_CCMR2)

Definition at line 3776 of file STM8AF_STM8S.h.

5.1.2.1285 _TIM3_OC2M0

```
#define _TIM3_OC2M0 ((uint8_t) (0x01 << 4))
```

TIM3 Output compare 2 mode [0] (in _TIM3_CCMR2)

Definition at line 3777 of file STM8AF_STM8S.h.

5.1.2.1286 _TIM3_OC2M1

```
#define _TIM3_OC2M1 ((uint8_t) (0x01 << 5))
```

TIM3 Output compare 2 mode [1] (in _TIM3_CCMR2)

Definition at line 3778 of file STM8AF_STM8S.h.

5.1.2.1287 _TIM3_OC2M2

```
#define _TIM3_OC2M2 ((uint8_t) (0x01 << 6))
```

TIM3 Output compare 2 mode [2] (in _TIM3_CCMR2)

Definition at line 3779 of file STM8AF_STM8S.h.

5.1.2.1288 _TIM3_OC2PE

```
#define _TIM3_OC2PE ((uint8_t) (0x01 << 3))
```

TIM3 Output compare 2 preload enable [0] (in _TIM3_CCMR2)

Definition at line 3775 of file STM8AF_STM8S.h.

5.1.2.1289 _TIM3_OPM

```
#define _TIM3_OPM ((uint8_t) (0x01 << 3))
```

TIM3 One-pulse mode [0] (in _TIM3_CR1)

Definition at line 3719 of file STM8AF_STM8S.h.

5.1.2.1290 _TIM3_PSC

```
#define _TIM3_PSC ((uint8_t) (0x0F << 0))
```

TIM3 clock prescaler [3:0] (in _TIM3_PSCR)

Definition at line 3802 of file STM8AF_STM8S.h.

5.1.2.1291 _TIM3_PSC0

```
#define _TIM3_PSC0 ((uint8_t) (0x01 << 0))
```

TIM3 clock prescaler [0] (in _TIM3_PSCR)

Definition at line 3803 of file STM8AF_STM8S.h.

5.1.2.1292 _TIM3_PSC1

```
#define _TIM3_PSC1 ((uint8_t) (0x01 << 1))
```

TIM3 clock prescaler [1] (in _TIM3_PSCR)

Definition at line 3804 of file STM8AF_STM8S.h.

5.1.2.1293 _TIM3_PSC2

```
#define _TIM3_PSC2 ((uint8_t) (0x01 << 2))
```

TIM3 clock prescaler [2] (in _TIM3_PSCR)

Definition at line 3805 of file STM8AF_STM8S.h.

5.1.2.1294 _TIM3_PSC3

```
#define _TIM3_PSC3 ((uint8_t) (0x01 << 3))
```

TIM3 clock prescaler [3] (in _TIM3_PSCR)

Definition at line 3806 of file STM8AF_STM8S.h.

5.1.2.1295 _TIM3_PSCR

```
#define _TIM3_PSCR _SFR(uint8_t, TIM3_AddressBase+0x0C)
```

TIM3 clock prescaler register.

Definition at line 3688 of file STM8AF_STM8S.h.

5.1.2.1296 _TIM3_PSCR_RESET_VALUE

```
#define _TIM3_PSCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 clock prescaler register reset value.

Definition at line 3707 of file STM8AF_STM8S.h.

5.1.2.1297 _TIM3_SR1

```
#define _TIM3_SR1 _SFR(uint8_t, TIM3_AddressBase+0x02)
```

TIM3 status register 1.

Definition at line 3680 of file STM8AF_STM8S.h.

5.1.2.1298 _TIM3_SR1_RESET_VALUE

```
#define _TIM3_SR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 status register 1 reset value.

Definition at line 3699 of file STM8AF_STM8S.h.

5.1.2.1299 _TIM3_SR2

```
#define _TIM3_SR2 _SFR(uint8_t, TIM3_AddressBase+0x03)
```

TIM3 status register 2.

Definition at line 3681 of file STM8AF_STM8S.h.

5.1.2.1300 _TIM3_SR2_RESET_VALUE

```
#define _TIM3_SR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 status register 2 reset value.

Definition at line 3700 of file STM8AF_STM8S.h.

5.1.2.1301 _TIM3_UDIS

```
#define _TIM3_UDIS ((uint8_t) (0x01 << 1))
```

TIM3 Update disable [0] (in _TIM3_CR1)

Definition at line 3717 of file STM8AF_STM8S.h.

5.1.2.1302 _TIM3_UG

```
#define _TIM3_UG ((uint8_t) (0x01 << 0))
```

TIM3 Update generation [0] (in _TIM3_EGR)

Definition at line 3742 of file STM8AF_STM8S.h.

5.1.2.1303 _TIM3_UIE

```
#define _TIM3_UIE ((uint8_t) (0x01 << 0))
```

TIM3 Update interrupt enable [0] (in _TIM3_IER)

Definition at line 3724 of file STM8AF_STM8S.h.

5.1.2.1304 _TIM3_UIF

```
#define _TIM3_UIF ((uint8_t) (0x01 << 0))
```

TIM3 Update interrupt flag [0] (in _TIM3_SR1)

Definition at line 3730 of file STM8AF_STM8S.h.

5.1.2.1305 _TIM3_URS

```
#define _TIM3_URS ((uint8_t) (0x01 << 2))
```

TIM3 Update request source [0] (in _TIM3_CR1)

Definition at line 3718 of file STM8AF_STM8S.h.

5.1.2.1306 _TIM4

```
#define _TIM4 _SFR(TIM4_t, TIM4_AddressBase)
```

TIM4 struct/bit access.

Definition at line 3879 of file STM8AF_STM8S.h.

5.1.2.1307 _TIM4_ARPE

```
#define _TIM4_ARPE ((uint8_t) (0x01 << 7))
```

TIM4 Auto-reload preload enable [0] (in _TIM4_CR)

Definition at line 3913 of file STM8AF_STM8S.h.

5.1.2.1308 _TIM4_ARR

```
#define _TIM4_ARR _SFR(uint8_t, TIM4_AddressBase+0x06)
```

TIM4 auto-reload register.

Definition at line 3895 of file STM8AF_STM8S.h.

5.1.2.1309 _TIM4_ARR_RESET_VALUE

```
#define _TIM4_ARR_RESET_VALUE ((uint8_t) 0xFF)
```

TIM4 auto-reload register reset value.

Definition at line 3905 of file STM8AF_STM8S.h.

5.1.2.1310 _TIM4_CEN

```
#define _TIM4_CEN ((uint8_t) (0x01 << 0))
```

TIM4 Counter enable [0] (in _TIM4_CR)

Definition at line 3908 of file STM8AF_STM8S.h.

5.1.2.1311 _TIM4_CNTR

```
#define _TIM4_CNTR _SFR(uint8_t, TIM4_AddressBase+0x04)
```

TIM4 counter register.

Definition at line 3893 of file STM8AF_STM8S.h.

5.1.2.1312 _TIM4_CNTR_RESET_VALUE

```
#define _TIM4_CNTR_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 counter register reset value.

Definition at line 3903 of file STM8AF_STM8S.h.

5.1.2.1313 _TIM4_CR

```
#define _TIM4_CR _SFR(uint8_t, TIM4_AddressBase+0x00)
```

TIM4 control register.

Definition at line 3880 of file STM8AF_STM8S.h.

5.1.2.1314 _TIM4_CR_RESET_VALUE

```
#define _TIM4_CR_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 control register reset value.

Definition at line 3899 of file STM8AF_STM8S.h.

5.1.2.1315 _TIM4_EGR

```
#define _TIM4_EGR _SFR(uint8_t, TIM4_AddressBase+0x03)
```

TIM4 event generation register.

Definition at line 3892 of file STM8AF_STM8S.h.

5.1.2.1316 _TIM4_EGR_RESET_VALUE

```
#define _TIM4_EGR_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 event generation register reset value.

Definition at line 3902 of file STM8AF_STM8S.h.

5.1.2.1317 _TIM4_IER

```
#define _TIM4_IER _SFR(uint8_t, TIM4_AddressBase+0x01)
```

TIM4 interrupt enable register.

Definition at line 3890 of file STM8AF_STM8S.h.

5.1.2.1318 _TIM4_IER_RESET_VALUE

```
#define _TIM4_IER_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 interrupt enable register reset value.

Definition at line 3900 of file STM8AF_STM8S.h.

5.1.2.1319 _TIM4_OPM

```
#define _TIM4_OPM ((uint8_t) (0x01 << 3))
```

TIM4 One-pulse mode [0] (in _TIM4_CR)

Definition at line 3911 of file STM8AF_STM8S.h.

5.1.2.1320 _TIM4_PSC

```
#define _TIM4_PSC ((uint8_t) (0x07 << 0))
```

TIM4 clock prescaler [2:0] (in _TIM4_PSCR)

Definition at line 3928 of file STM8AF_STM8S.h.

5.1.2.1321 _TIM4_PSC0

```
#define _TIM4_PSC0 ((uint8_t) (0x01 << 0))
```

TIM4 clock prescaler [0] (in _TIM4_PSCR)

Definition at line 3929 of file STM8AF_STM8S.h.

5.1.2.1322 _TIM4_PSC1

```
#define _TIM4_PSC1 ((uint8_t) (0x01 << 1))
```

TIM4 clock prescaler [1] (in _TIM4_PSCR)

Definition at line 3930 of file STM8AF_STM8S.h.

5.1.2.1323 _TIM4_PSC2

```
#define _TIM4_PSC2 ((uint8_t) (0x01 << 2))
```

TIM4 clock prescaler [2] (in _TIM4_PSCR)

Definition at line 3931 of file STM8AF_STM8S.h.

5.1.2.1324 _TIM4_PSCR

```
#define _TIM4_PSCR _SFR(uint8_t, TIM4_AddressBase+0x05)
```

TIM4 clock prescaler register.

Definition at line 3894 of file STM8AF_STM8S.h.

5.1.2.1325 _TIM4_PSCR_RESET_VALUE

```
#define _TIM4_PSCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 clock prescaler register reset value.

Definition at line 3904 of file STM8AF_STM8S.h.

5.1.2.1326 _TIM4_SR

```
#define _TIM4_SR _SFR(uint8_t, TIM4_AddressBase+0x02)
```

TIM4 status register.

Definition at line 3891 of file STM8AF_STM8S.h.

5.1.2.1327 _TIM4_SR_RESET_VALUE

```
#define _TIM4_SR_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 status register reset value.

Definition at line 3901 of file STM8AF_STM8S.h.

5.1.2.1328 _TIM4_UDIS

```
#define _TIM4_UDIS ((uint8_t) (0x01 << 1))
```

TIM4 Update disable [0] (in _TIM4_CR)

Definition at line 3909 of file STM8AF_STM8S.h.

5.1.2.1329 _TIM4_UG

```
#define _TIM4_UG ((uint8_t) (0x01 << 0))
```

TIM4 Update generation [0] (in _TIM4_EGR)

Definition at line 3924 of file STM8AF_STM8S.h.

5.1.2.1330 _TIM4_UIE

```
#define _TIM4_UIE ((uint8_t) (0x01 << 0))
```

TIM4 Update interrupt enable [0] (in _TIM4_IER)

Definition at line 3916 of file STM8AF_STM8S.h.

5.1.2.1331 _TIM4_UIF

```
#define _TIM4_UIF ((uint8_t) (0x01 << 0))
```

TIM4 Update interrupt flag [0] (in _TIM4_SR)

Definition at line 3920 of file STM8AF_STM8S.h.

5.1.2.1332 _TIM4_URS

```
#define _TIM4_URS ((uint8_t) (0x01 << 2))
```

TIM4 Update request source [0] (in _TIM4_CR)

Definition at line 3910 of file STM8AF_STM8S.h.

5.1.2.1333 _TIM5

```
#define _TIM5 _SFR(TIM5_t, TIM5_AddressBase)
```

TIM5 struct/bit access.

Definition at line 4177 of file STM8AF_STM8S.h.

5.1.2.1334 _TIM5_ARPE

```
#define _TIM5_ARPE ((uint8_t) (0x01 << 7))
```

TIM5 Auto-reload preload enable [0] (in _TIM5_CR1)

Definition at line 4233 of file STM8AF_STM8S.h.

5.1.2.1335 _TIM5_ARRH

```
#define _TIM5_ARRH _SFR(uint8_t, TIM5_AddressBase+0x0F)
```

TIM5 auto-reload register high byte.

Definition at line 4193 of file STM8AF_STM8S.h.

5.1.2.1336 _TIM5_ARRH_RESET_VALUE

```
#define _TIM5_ARRH_RESET_VALUE ((uint8_t) 0xFF)
```

TIM5 auto-reload register high byte reset value.

Definition at line 4218 of file STM8AF_STM8S.h.

5.1.2.1337 _TIM5_ARRL

```
#define _TIM5_ARRL _SFR(uint8_t, TIM5_AddressBase+0x10)
```

TIM5 auto-reload register low byte.

Definition at line 4194 of file STM8AF_STM8S.h.

5.1.2.1338 _TIM5_ARRL_RESET_VALUE

```
#define _TIM5_ARRL_RESET_VALUE ((uint8_t) 0xFF)
```

TIM5 auto-reload register low byte reset value.

Definition at line 4219 of file STM8AF_STM8S.h.

5.1.2.1339 _TIM5_CC1E

```
#define _TIM5_CC1E ((uint8_t) (0x01 << 0))
```

TIM5 Capture/compare 1 output enable [0] (in _TIM5_CCER1)

Definition at line 4362 of file STM8AF_STM8S.h.

5.1.2.1340 _TIM5_CC1G

```
#define _TIM5_CC1G ((uint8_t) (0x01 << 1))
```

TIM5 Capture/compare 1 generation [0] (in _TIM5_EGR)

Definition at line 4285 of file STM8AF_STM8S.h.

5.1.2.1341 _TIM5_CC1IE

```
#define _TIM5_CC1IE ((uint8_t) (0x01 << 1))
```

TIM5 Capture/compare 1 interrupt enable [0] (in _TIM5_IER)

Definition at line 4260 of file STM8AF_STM8S.h.

5.1.2.1342 _TIM5_CC1IF

```
#define _TIM5_CC1IF ((uint8_t) (0x01 << 1))
```

TIM5 Capture/compare 1 interrupt flag [0] (in _TIM5_SR1)

Definition at line 4269 of file STM8AF_STM8S.h.

5.1.2.1343 _TIM5_CC1OF

```
#define _TIM5_CC1OF ((uint8_t) (0x01 << 1))
```

TIM5 Capture/compare 1 overcapture flag [0] (in _TIM5_SR2)

Definition at line 4278 of file STM8AF_STM8S.h.

5.1.2.1344 _TIM5_CC1P

```
#define _TIM5_CC1P ((uint8_t) (0x01 << 1))
```

TIM5 Capture/compare 1 output polarity [0] (in _TIM5_CCER1)

Definition at line 4363 of file STM8AF_STM8S.h.

5.1.2.1345 _TIM5_CC1S

```
#define _TIM5_CC1S ((uint8_t) (0x03 << 0))
```

TIM5 Compare 1 selection [1:0] (in _TIM5_CCMR1)

Definition at line 4293 of file STM8AF_STM8S.h.

5.1.2.1346 _TIM5_CC1S0

```
#define _TIM5_CC1S0 ((uint8_t) (0x01 << 0))
```

TIM5 Compare 1 selection [0] (in _TIM5_CCMR1)

Definition at line 4294 of file STM8AF_STM8S.h.

5.1.2.1347 _TIM5_CC1S1

```
#define _TIM5_CC1S1 ((uint8_t) (0x01 << 1))
```

TIM5 Compare 1 selection [1] (in _TIM5_CCMR1)

Definition at line 4295 of file STM8AF_STM8S.h.

5.1.2.1348 _TIM5_CC2E

```
#define _TIM5_CC2E ((uint8_t) (0x01 << 4))
```

TIM5 Capture/compare 2 output enable [0] (in _TIM5_CCER1)

Definition at line 4365 of file STM8AF_STM8S.h.

5.1.2.1349 _TIM5_CC2G

```
#define _TIM5_CC2G ((uint8_t) (0x01 << 2))
```

TIM5 Capture/compare 2 generation [0] (in _TIM5_EGR)

Definition at line 4286 of file STM8AF_STM8S.h.

5.1.2.1350 _TIM5_CC2IE

```
#define _TIM5_CC2IE ((uint8_t) (0x01 << 2))
```

TIM5 Capture/compare 2 interrupt enable [0] (in _TIM5_IER)

Definition at line 4261 of file STM8AF_STM8S.h.

5.1.2.1351 _TIM5_CC2IF

```
#define _TIM5_CC2IF ((uint8_t) (0x01 << 2))
```

TIM5 Capture/compare 2 interrupt flag [0] (in _TIM5_SR1)

Definition at line 4270 of file STM8AF_STM8S.h.

5.1.2.1352 _TIM5_CC2OF

```
#define _TIM5_CC2OF ((uint8_t) (0x01 << 2))
```

TIM5 Capture/compare 2 overcapture flag [0] (in _TIM5_SR2)

Definition at line 4279 of file STM8AF_STM8S.h.

5.1.2.1353 _TIM5_CC2P

```
#define _TIM5_CC2P ((uint8_t) (0x01 << 5))
```

TIM5 Capture/compare 2 output polarity [0] (in _TIM5_CCER1)

Definition at line 4366 of file STM8AF_STM8S.h.

5.1.2.1354 _TIM5_CC2S

```
#define _TIM5_CC2S ((uint8_t) (0x03 << 0))
```

TIM5 Compare 2 selection [1:0] (in _TIM5_CCMR2)

Definition at line 4316 of file STM8AF_STM8S.h.

5.1.2.1355 _TIM5_CC2S0

```
#define _TIM5_CC2S0 ((uint8_t) (0x01 << 0))
```

TIM5 Compare 2 selection [0] (in _TIM5_CCMR2)

Definition at line 4317 of file STM8AF_STM8S.h.

5.1.2.1356 _TIM5_CC2S1

```
#define _TIM5_CC2S1 ((uint8_t) (0x01 << 1))
```

TIM5 Compare 2 selection [1] (in _TIM5_CCMR2)

Definition at line 4318 of file STM8AF_STM8S.h.

5.1.2.1357 _TIM5_CC3E

```
#define _TIM5_CC3E ((uint8_t) (0x01 << 0))
```

TIM5 Capture/compare 3 output enable [0] (in _TIM5_CCER2)

Definition at line 4370 of file STM8AF_STM8S.h.

5.1.2.1358 _TIM5_CC3G

```
#define _TIM5_CC3G ((uint8_t) (0x01 << 3))
```

TIM5 Capture/compare 3 generation [0] (in _TIM5_EGR)

Definition at line 4287 of file STM8AF_STM8S.h.

5.1.2.1359 _TIM5_CC3IE

```
#define _TIM5_CC3IE ((uint8_t) (0x01 << 3))
```

TIM5 Capture/compare 3 interrupt enable [0] (in _TIM5_IER)

Definition at line 4262 of file STM8AF_STM8S.h.

5.1.2.1360 _TIM5_CC3IF

```
#define _TIM5_CC3IF ((uint8_t) (0x01 << 3))
```

TIM5 Capture/compare 3 interrupt flag [0] (in _TIM5_SR1)

Definition at line 4271 of file STM8AF_STM8S.h.

5.1.2.1361 _TIM5_CC3OF

```
#define _TIM5_CC3OF ((uint8_t) (0x01 << 3))
```

TIM5 Capture/compare 3 overcapture flag [0] (in _TIM5_SR2)

Definition at line 4280 of file STM8AF_STM8S.h.

5.1.2.1362 _TIM5_CC3P

```
#define _TIM5_CC3P ((uint8_t) (0x01 << 1))
```

TIM5 Capture/compare 3 output polarity [0] (in _TIM5_CCER2)

Definition at line 4371 of file STM8AF_STM8S.h.

5.1.2.1363 _TIM5_CC3S

```
#define _TIM5_CC3S ((uint8_t) (0x03 << 0))
```

TIM5 Compare 3 selection [1:0] (in _TIM5_CCMR3)

Definition at line 4339 of file STM8AF_STM8S.h.

5.1.2.1364 _TIM5_CC3S0

```
#define _TIM5_CC3S0 ((uint8_t) (0x01 << 0))
```

TIM5 Compare 3 selection [0] (in _TIM5_CCMR3)

Definition at line 4340 of file STM8AF_STM8S.h.

5.1.2.1365 _TIM5_CC3S1

```
#define _TIM5_CC3S1 ((uint8_t) (0x01 << 1))
```

TIM5 Compare 3 selection [1] (in _TIM5_CCMR3)

Definition at line 4341 of file STM8AF_STM8S.h.

5.1.2.1366 _TIM5_CCER1

```
#define _TIM5_CCER1 _SFR(uint8_t, TIM5_AddressBase+0x0A)
```

TIM5 Capture/compare enable register 1.

Definition at line 4188 of file STM8AF_STM8S.h.

5.1.2.1367 _TIM5_CCER1_RESET_VALUE

```
#define _TIM5_CCER1_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 Capture/compare enable register 1 reset value.

Definition at line 4213 of file STM8AF_STM8S.h.

5.1.2.1368 _TIM5_CCER2

```
#define _TIM5_CCER2 _SFR(uint8_t, TIM5_AddressBase+0x0B)
```

TIM5 Capture/compare enable register 2.

Definition at line 4189 of file STM8AF_STM8S.h.

5.1.2.1369 _TIM5_CCER2_RESET_VALUE

```
#define _TIM5_CCER2_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 Capture/compare enable register 2 reset value.

Definition at line 4214 of file STM8AF_STM8S.h.

5.1.2.1370 _TIM5_CCMR1

```
#define _TIM5_CCMR1 _SFR(uint8_t, TIM5_AddressBase+0x07)
```

TIM5 Capture/compare mode register 1.

Definition at line 4185 of file STM8AF_STM8S.h.

5.1.2.1371 _TIM5_CCMR1_RESET_VALUE

```
#define _TIM5_CCMR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 Capture/compare mode register 1 reset value.

Definition at line 4210 of file STM8AF_STM8S.h.

5.1.2.1372 _TIM5_CCMR2

```
#define _TIM5_CCMR2 _SFR(uint8_t, TIM5_AddressBase+0x08)
```

TIM5 Capture/compare mode register 2.

Definition at line 4186 of file STM8AF_STM8S.h.

5.1.2.1373 _TIM5_CCMR2_RESET_VALUE

```
#define _TIM5_CCMR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 Capture/compare mode register 2 reset value.

Definition at line 4211 of file STM8AF_STM8S.h.

5.1.2.1374 _TIM5_CCMR3

```
#define _TIM5_CCMR3 _SFR(uint8_t, TIM5_AddressBase+0x09)
```

TIM5 Capture/compare mode register 3.

Definition at line 4187 of file STM8AF_STM8S.h.

5.1.2.1375 _TIM5_CCMR3_RESET_VALUE

```
#define _TIM5_CCMR3_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 Capture/compare mode register 3 reset value.

Definition at line 4212 of file STM8AF_STM8S.h.

5.1.2.1376 _TIM5_CCPC

```
#define _TIM5_CCPC ((uint8_t) (0x01 << 0))
```

TIM5 Capture/compare preloaded control [0] (in _TIM5_CR2)

Definition at line 4236 of file STM8AF_STM8S.h.

5.1.2.1377 _TIM5_CCR1H

```
#define _TIM5_CCR1H _SFR(uint8_t, TIM5_AddressBase+0x11)
```

TIM5 16-bit capture/compare value 1 high byte.

Definition at line 4195 of file STM8AF_STM8S.h.

5.1.2.1378 _TIM5_CCR1H_RESET_VALUE

```
#define _TIM5_CCR1H_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 16-bit capture/compare value 1 high byte reset value.

Definition at line 4220 of file STM8AF_STM8S.h.

5.1.2.1379 _TIM5_CCR1L

```
#define _TIM5_CCR1L _SFR(uint8_t, TIM5_AddressBase+0x12)
```

TIM5 16-bit capture/compare value 1 low byte.

Definition at line 4196 of file STM8AF_STM8S.h.

5.1.2.1380 _TIM5_CCR1L_RESET_VALUE

```
#define _TIM5_CCR1L_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 16-bit capture/compare value 1 low byte reset value.

Definition at line 4221 of file STM8AF_STM8S.h.

5.1.2.1381 _TIM5_CCR2H

```
#define _TIM5_CCR2H _SFR(uint8_t, TIM5_AddressBase+0x13)
```

TIM5 16-bit capture/compare value 2 high byte.

Definition at line 4197 of file STM8AF_STM8S.h.

5.1.2.1382 _TIM5_CCR2H_RESET_VALUE

```
#define _TIM5_CCR2H_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 16-bit capture/compare value 2 high byte reset value.

Definition at line 4222 of file STM8AF_STM8S.h.

5.1.2.1383 _TIM5_CCR2L

```
#define _TIM5_CCR2L _SFR(uint8_t, TIM5_AddressBase+0x14)
```

TIM5 16-bit capture/compare value 2 low byte.

Definition at line 4198 of file STM8AF_STM8S.h.

5.1.2.1384 _TIM5_CCR2L_RESET_VALUE

```
#define _TIM5_CCR2L_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 16-bit capture/compare value 2 low byte reset value.

Definition at line 4223 of file STM8AF_STM8S.h.

5.1.2.1385 _TIM5_CCR3H

```
#define _TIM5_CCR3H _SFR(uint8_t, TIM5_AddressBase+0x15)
```

TIM5 16-bit capture/compare value 3 high byte.

Definition at line 4199 of file STM8AF_STM8S.h.

5.1.2.1386 _TIM5_CCR3H_RESET_VALUE

```
#define _TIM5_CCR3H_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 16-bit capture/compare value 3 high byte reset value.

Definition at line 4224 of file STM8AF_STM8S.h.

5.1.2.1387 _TIM5_CCR3L

```
#define _TIM5_CCR3L _SFR(uint8_t, TIM5_AddressBase+0x16)
```

TIM5 16-bit capture/compare value 3 low byte.

Definition at line 4200 of file STM8AF_STM8S.h.

5.1.2.1388 _TIM5_CCR3L_RESET_VALUE

```
#define _TIM5_CCR3L_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 16-bit capture/compare value 3 low byte reset value.

Definition at line 4225 of file STM8AF_STM8S.h.

5.1.2.1389 _TIM5_CEN

```
#define _TIM5_CEN ((uint8_t) (0x01 << 0))
```

TIM5 Counter enable [0] (in _TIM5_CR1)

Definition at line 4228 of file STM8AF_STM8S.h.

5.1.2.1390 _TIM5_CNTRH

```
#define _TIM5_CNTRH _SFR(uint8_t, TIM5_AddressBase+0x0C)
```

TIM5 counter register high byte.

Definition at line 4190 of file STM8AF_STM8S.h.

5.1.2.1391 _TIM5_CNTRH_RESET_VALUE

```
#define _TIM5_CNTRH_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 counter register high byte reset value.

Definition at line 4215 of file STM8AF_STM8S.h.

5.1.2.1392 _TIM5_CNTRL

```
#define _TIM5_CNTRL _SFR(uint8_t, TIM5_AddressBase+0x0D)
```

TIM5 counter register low byte.

Definition at line 4191 of file STM8AF_STM8S.h.

5.1.2.1393 _TIM5_CNTRL_RESET_VALUE

```
#define _TIM5_CNTRL_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 counter register low byte reset value.

Definition at line 4216 of file STM8AF_STM8S.h.

5.1.2.1394 _TIM5_COMS

```
#define _TIM5_COMS ((uint8_t) (0x01 << 2))
```

TIM5 Capture/compare control update selection [0] (in _TIM5_CR2)

Definition at line 4238 of file STM8AF_STM8S.h.

5.1.2.1395 _TIM5_CR1

```
#define _TIM5_CR1 _SFR(uint8_t, TIM5_AddressBase+0x00)
```

TIM5 control register 1.

Definition at line 4178 of file STM8AF_STM8S.h.

5.1.2.1396 _TIM5_CR1_RESET_VALUE

```
#define _TIM5_CR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 control register 1 reset value.

Definition at line 4203 of file STM8AF_STM8S.h.

5.1.2.1397 _TIM5_CR2

```
#define _TIM5_CR2 _SFR(uint8_t, TIM5_AddressBase+0x01)
```

TIM5 control register 2.

Definition at line 4179 of file STM8AF_STM8S.h.

5.1.2.1398 _TIM5_CR2_RESET_VALUE

```
#define _TIM5_CR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 control register 2 reset value.

Definition at line 4204 of file STM8AF_STM8S.h.

5.1.2.1399 _TIM5_EGR

```
#define _TIM5_EGR _SFR(uint8_t, TIM5_AddressBase+0x06)
```

TIM5 Event generation register.

Definition at line 4184 of file STM8AF_STM8S.h.

5.1.2.1400 _TIM5_EGR_RESET_VALUE

```
#define _TIM5_EGR_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 Event generation register reset value.

Definition at line 4209 of file STM8AF_STM8S.h.

5.1.2.1401 _TIM5_IC1F

```
#define _TIM5_IC1F ((uint8_t) (0x0F << 4))
```

TIM5 Output compare 1 mode [3:0] (in _TIM5_CCMR1)

Definition at line 4309 of file STM8AF_STM8S.h.

5.1.2.1402 _TIM5_IC1F0

```
#define _TIM5_IC1F0 ((uint8_t) (0x01 << 4))
```

TIM5 Input capture 1 filter [0] (in _TIM5_CCMR1)

Definition at line 4310 of file STM8AF_STM8S.h.

5.1.2.1403 _TIM5_IC1F1

```
#define _TIM5_IC1F1 ((uint8_t) (0x01 << 5))
```

TIM5 Input capture 1 filter [1] (in _TIM5_CCMR1)

Definition at line 4311 of file STM8AF_STM8S.h.

5.1.2.1404 _TIM5_IC1F2

```
#define _TIM5_IC1F2 ((uint8_t) (0x01 << 6))
```

TIM5 Input capture 1 filter [2] (in _TIM5_CCMR1)

Definition at line 4312 of file STM8AF_STM8S.h.

5.1.2.1405 _TIM5_IC1F3

```
#define _TIM5_IC1F3 ((uint8_t) (0x01 << 7))
```

TIM5 Input capture 1 filter [3] (in _TIM5_CCMR1)

Definition at line 4313 of file STM8AF_STM8S.h.

5.1.2.1406 _TIM5_IC1PSC

```
#define _TIM5_IC1PSC ((uint8_t) (0x03 << 2))
```

TIM5 Input capture 1 prescaler [1:0] (in _TIM5_CCMR1)

Definition at line 4306 of file STM8AF_STM8S.h.

5.1.2.1407 _TIM5_IC1PSC0

```
#define _TIM5_IC1PSC0 ((uint8_t) (0x01 << 2))
```

TIM5 Input capture 1 prescaler [0] (in _TIM5_CCMR1)

Definition at line 4307 of file STM8AF_STM8S.h.

5.1.2.1408 _TIM5_IC1PSC1

```
#define _TIM5_IC1PSC1 ((uint8_t) (0x01 << 3))
```

TIM5 Input capture 1 prescaler [1] (in _TIM5_CCMR1)

Definition at line 4308 of file STM8AF_STM8S.h.

5.1.2.1409 _TIM5_IC2F

```
#define _TIM5_IC2F ((uint8_t) (0x0F << 4))
```

TIM5 Output compare 2 mode [3:0] (in _TIM5_CCMR2)

Definition at line 4332 of file STM8AF_STM8S.h.

5.1.2.1410 _TIM5_IC2F0

```
#define _TIM5_IC2F0 ((uint8_t) (0x01 << 4))
```

TIM5 Input capture 2 filter [0] (in _TIM5_CCMR2)

Definition at line 4333 of file STM8AF_STM8S.h.

5.1.2.1411 _TIM5_IC2F1

```
#define _TIM5_IC2F1 ((uint8_t) (0x01 << 5))
```

TIM5 Input capture 2 filter [1] (in _TIM5_CCMR2)

Definition at line 4334 of file STM8AF_STM8S.h.

5.1.2.1412 _TIM5_IC2F2

```
#define _TIM5_IC2F2 ((uint8_t) (0x01 << 6))
```

TIM5 Input capture 2 filter [2] (in _TIM5_CCMR2)

Definition at line 4335 of file STM8AF_STM8S.h.

5.1.2.1413 _TIM5_IC2F3

```
#define _TIM5_IC2F3 ((uint8_t) (0x01 << 7))
```

TIM5 Input capture 2 filter [3] (in _TIM5_CCMR2)

Definition at line 4336 of file STM8AF_STM8S.h.

5.1.2.1414 _TIM5_IC2PSC

```
#define _TIM5_IC2PSC ((uint8_t) (0x03 << 2))
```

TIM5 Input capture 2 prescaler [1:0] (in _TIM5_CCMR2)

Definition at line 4329 of file STM8AF_STM8S.h.

5.1.2.1415 _TIM5_IC2PSC0

```
#define _TIM5_IC2PSC0 ((uint8_t) (0x01 << 2))
```

TIM5 Input capture 2 prescaler [0] (in _TIM5_CCMR2)

Definition at line 4330 of file STM8AF_STM8S.h.

5.1.2.1416 _TIM5_IC2PSC1

```
#define _TIM5_IC2PSC1 ((uint8_t) (0x01 << 3))
```

TIM5 Input capture 2 prescaler [1] (in _TIM5_CCMR2)

Definition at line 4331 of file STM8AF_STM8S.h.

5.1.2.1417 _TIM5_IC3F

```
#define _TIM5_IC3F ((uint8_t) (0x0F << 4))
```

TIM5 Output compare 3 mode [3:0] (in _TIM5_CCMR3)

Definition at line 4355 of file STM8AF_STM8S.h.

5.1.2.1418 _TIM5_IC3F0

```
#define _TIM5_IC3F0 ((uint8_t) (0x01 << 4))
```

TIM5 Input capture 3 filter [0] (in _TIM5_CCMR3)

Definition at line 4356 of file STM8AF_STM8S.h.

5.1.2.1419 _TIM5_IC3F1

```
#define _TIM5_IC3F1 ((uint8_t) (0x01 << 5))
```

TIM5 Input capture 3 filter [1] (in _TIM5_CCMR3)

Definition at line 4357 of file STM8AF_STM8S.h.

5.1.2.1420 _TIM5_IC3F2

```
#define _TIM5_IC3F2 ((uint8_t) (0x01 << 6))
```

TIM5 Input capture 3 filter [2] (in _TIM5_CCMR3)

Definition at line 4358 of file STM8AF_STM8S.h.

5.1.2.1421 _TIM5_IC3F3

```
#define _TIM5_IC3F3 ((uint8_t) (0x01 << 7))
```

TIM5 Input capture 3 filter [3] (in _TIM5_CCMR3)

Definition at line 4359 of file STM8AF_STM8S.h.

5.1.2.1422 _TIM5_IC3PSC

```
#define _TIM5_IC3PSC ((uint8_t) (0x03 << 2))
```

TIM5 Input capture 3 prescaler [1:0] (in _TIM5_CCMR3)

Definition at line 4352 of file STM8AF_STM8S.h.

5.1.2.1423 _TIM5_IC3PSC0

```
#define _TIM5_IC3PSC0 ((uint8_t) (0x01 << 2))
```

TIM5 Input capture 3 prescaler [0] (in _TIM5_CCMR3)

Definition at line 4353 of file STM8AF_STM8S.h.

5.1.2.1424 _TIM5_IC3PSC1

```
#define _TIM5_IC3PSC1 ((uint8_t) (0x01 << 3))
```

TIM5 Input capture 3 prescaler [1] (in _TIM5_CCMR3)

Definition at line 4354 of file STM8AF_STM8S.h.

5.1.2.1425 _TIM5_IER

```
#define _TIM5_IER _SFR(uint8_t, TIM5_AddressBase+0x03)
```

TIM5 interrupt enable register.

Definition at line 4181 of file STM8AF_STM8S.h.

5.1.2.1426 _TIM5_IER_RESET_VALUE

```
#define _TIM5_IER_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 interrupt enable register reset value.

Definition at line 4206 of file STM8AF_STM8S.h.

5.1.2.1427 _TIM5_MMS

```
#define _TIM5_MMS ((uint8_t) (0x07 << 4))
```

TIM5 Master mode selection [2:0] (in _TIM5_CR2)

Definition at line 4240 of file STM8AF_STM8S.h.

5.1.2.1428 _TIM5_MMS0

```
#define _TIM5_MMS0 ((uint8_t) (0x01 << 4))
```

TIM5 Master mode selection [0] (in _TIM5_CR2)

Definition at line 4241 of file STM8AF_STM8S.h.

5.1.2.1429 _TIM5_MMS1

```
#define _TIM5_MMS1 ((uint8_t) (0x01 << 5))
```

TIM5 Master mode selection [1] (in _TIM5_CR2)

Definition at line 4242 of file STM8AF_STM8S.h.

5.1.2.1430 _TIM5_MMS2

```
#define _TIM5_MMS2 ((uint8_t) (0x01 << 6))
```

TIM5 Master mode selection [2] (in _TIM5_CR2)

Definition at line 4243 of file STM8AF_STM8S.h.

5.1.2.1431 _TIM5_MSM

```
#define _TIM5_MSM ((uint8_t) (0x01 << 7))
```

TIM5 Master/slave mode [0] (in _TIM5_SMCR)

Definition at line 4256 of file STM8AF_STM8S.h.

5.1.2.1432 _TIM5_OC1M

```
#define _TIM5_OC1M ((uint8_t) (0x07 << 4))
```

TIM5 Output compare 1 mode [2:0] (in _TIM5_CCMR1)

Definition at line 4298 of file STM8AF_STM8S.h.

5.1.2.1433 _TIM5_OC1M0

```
#define _TIM5_OC1M0 ((uint8_t) (0x01 << 4))
```

TIM5 Output compare 1 mode [0] (in _TIM5_CCMR1)

Definition at line 4299 of file STM8AF_STM8S.h.

5.1.2.1434 _TIM5_OC1M1

```
#define _TIM5_OC1M1 ((uint8_t) (0x01 << 5))
```

TIM5 Output compare 1 mode [1] (in _TIM5_CCMR1)

Definition at line 4300 of file STM8AF_STM8S.h.

5.1.2.1435 _TIM5_OC1M2

```
#define _TIM5_OC1M2 ((uint8_t) (0x01 << 6))
```

TIM5 Output compare 1 mode [2] (in _TIM5_CCMR1)

Definition at line 4301 of file STM8AF_STM8S.h.

5.1.2.1436 _TIM5_OC1PE

```
#define _TIM5_OC1PE ((uint8_t) (0x01 << 3))
```

TIM5 Output compare 1 preload enable [0] (in _TIM5_CCMR1)

Definition at line 4297 of file STM8AF_STM8S.h.

5.1.2.1437 _TIM5_OC2M

```
#define _TIM5_OC2M ((uint8_t) (0x07 << 4))
```

TIM5 Output compare 2 mode [2:0] (in _TIM5_CCMR2)

Definition at line 4321 of file STM8AF_STM8S.h.

5.1.2.1438 _TIM5_OC2M0

```
#define _TIM5_OC2M0 ((uint8_t) (0x01 << 4))
```

TIM5 Output compare 2 mode [0] (in _TIM5_CCMR2)

Definition at line 4322 of file STM8AF_STM8S.h.

5.1.2.1439 _TIM5_OC2M1

```
#define _TIM5_OC2M1 ((uint8_t) (0x01 << 5))
```

TIM5 Output compare 2 mode [1] (in _TIM5_CCMR2)

Definition at line 4323 of file STM8AF_STM8S.h.

5.1.2.1440 _TIM5_OC2M2

```
#define _TIM5_OC2M2 ((uint8_t) (0x01 << 6))
```

TIM5 Output compare 2 mode [2] (in _TIM5_CCMR2)

Definition at line 4324 of file STM8AF_STM8S.h.

5.1.2.1441 _TIM5_OC2PE

```
#define _TIM5_OC2PE ((uint8_t) (0x01 << 3))
```

TIM5 Output compare 2 preload enable [0] (in _TIM5_CCMR2)

Definition at line 4320 of file STM8AF_STM8S.h.

5.1.2.1442 _TIM5_OC3M

```
#define _TIM5_OC3M ((uint8_t) (0x07 << 4))
```

TIM5 Output compare 3 mode [2:0] (in _TIM5_CCMR3)

Definition at line 4344 of file STM8AF_STM8S.h.

5.1.2.1443 _TIM5_OC3M0

```
#define _TIM5_OC3M0 ((uint8_t) (0x01 << 4))
```

TIM5 Output compare 3 mode [0] (in _TIM5_CCMR3)

Definition at line 4345 of file STM8AF_STM8S.h.

5.1.2.1444 _TIM5_OC3M1

```
#define _TIM5_OC3M1 ((uint8_t) (0x01 << 5))
```

TIM5 Output compare 3 mode [1] (in _TIM5_CCMR3)

Definition at line 4346 of file STM8AF_STM8S.h.

5.1.2.1445 _TIM5_OC3M2

```
#define _TIM5_OC3M2 ((uint8_t) (0x01 << 6))
```

TIM5 Output compare 3 mode [2] (in _TIM5_CCMR3)

Definition at line 4347 of file STM8AF_STM8S.h.

5.1.2.1446 _TIM5_OC3PE

```
#define _TIM5_OC3PE ((uint8_t) (0x01 << 3))
```

TIM5 Output compare 3 preload enable [0] (in _TIM5_CCMR3)

Definition at line 4343 of file STM8AF_STM8S.h.

5.1.2.1447 _TIM5_OPM

```
#define _TIM5_OPM ((uint8_t) (0x01 << 3))
```

TIM5 One-pulse mode [0] (in _TIM5_CR1)

Definition at line 4231 of file STM8AF_STM8S.h.

5.1.2.1448 _TIM5_PSC

```
#define _TIM5_PSC ((uint8_t) (0x0F << 0))
```

TIM5 clock prescaler [3:0] (in _TIM5_PSCR)

Definition at line 4375 of file STM8AF_STM8S.h.

5.1.2.1449 _TIM5_PSC0

```
#define _TIM5_PSC0 ((uint8_t) (0x01 << 0))
```

TIM5 clock prescaler [0] (in _TIM5_PSCR)

Definition at line 4376 of file STM8AF_STM8S.h.

5.1.2.1450 _TIM5_PSC1

```
#define _TIM5_PSC1 ((uint8_t) (0x01 << 1))
```

TIM5 clock prescaler [1] (in _TIM5_PSCR)

Definition at line 4377 of file STM8AF_STM8S.h.

5.1.2.1451 _TIM5_PSC2

```
#define _TIM5_PSC2 ((uint8_t) (0x01 << 2))
```

TIM5 clock prescaler [2] (in _TIM5_PSCR)

Definition at line 4378 of file STM8AF_STM8S.h.

5.1.2.1452 _TIM5_PSC3

```
#define _TIM5_PSC3 ((uint8_t) (0x01 << 3))
```

TIM5 clock prescaler [3] (in _TIM5_PSCR)

Definition at line 4379 of file STM8AF_STM8S.h.

5.1.2.1453 _TIM5_PSCR

```
#define _TIM5_PSCR _SFR(uint8_t, TIM5_AddressBase+0x0E)
```

TIM5 clock prescaler register.

Definition at line 4192 of file STM8AF_STM8S.h.

5.1.2.1454 _TIM5_PSCR_RESET_VALUE

```
#define _TIM5_PSCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 clock prescaler register reset value.

Definition at line 4217 of file STM8AF_STM8S.h.

5.1.2.1455 _TIM5_SMCR

```
#define _TIM5_SMCR _SFR(uint8_t, TIM5_AddressBase+0x02)
```

TIM5 Slave mode control register.

Definition at line 4180 of file STM8AF_STM8S.h.

5.1.2.1456 _TIM5_SMCR_RESET_VALUE

```
#define _TIM5_SMCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 Slave mode control register reset value.

Definition at line 4205 of file STM8AF_STM8S.h.

5.1.2.1457 _TIM5_SMS

```
#define _TIM5_SMS ((uint8_t) (0x07 << 0))
```

TIM5 Clock/trigger/slave mode selection [2:0] (in _TIM5_SMCR)

Definition at line 4247 of file STM8AF_STM8S.h.

5.1.2.1458 _TIM5_SMS0

```
#define _TIM5_SMS0 ((uint8_t) (0x01 << 0))
```

TIM5 Clock/trigger/slave mode selection [0] (in _TIM5_SMCR)

Definition at line 4248 of file STM8AF_STM8S.h.

5.1.2.1459 _TIM5_SMS1

```
#define _TIM5_SMS1 ((uint8_t) (0x01 << 1))
```

TIM5 Clock/trigger/slave mode selection [1] (in _TIM5_SMCR)

Definition at line 4249 of file STM8AF_STM8S.h.

5.1.2.1460 _TIM5_SMS2

```
#define _TIM5_SMS2 ((uint8_t) (0x01 << 2))
```

TIM5 Clock/trigger/slave mode selection [2] (in _TIM5_SMCR)

Definition at line 4250 of file STM8AF_STM8S.h.

5.1.2.1461 _TIM5_SR1

```
#define _TIM5_SR1 _SFR(uint8_t, TIM5_AddressBase+0x04)
```

TIM5 status register 1.

Definition at line 4182 of file STM8AF_STM8S.h.

5.1.2.1462 _TIM5_SR1_RESET_VALUE

```
#define _TIM5_SR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 status register 1 reset value.

Definition at line 4207 of file STM8AF_STM8S.h.

5.1.2.1463 _TIM5_SR2

```
#define _TIM5_SR2 _SFR(uint8_t, TIM5_AddressBase+0x05)
```

TIM5 status register 2.

Definition at line 4183 of file STM8AF_STM8S.h.

5.1.2.1464 _TIM5_SR2_RESET_VALUE

```
#define _TIM5_SR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM5 status register 2 reset value.

Definition at line 4208 of file STM8AF_STM8S.h.

5.1.2.1465 _TIM5_TG

```
#define _TIM5_TG ((uint8_t) (0x01 << 6))
```

TIM5 Trigger generation [0] (in _TIM5_EGR)

Definition at line 4289 of file STM8AF_STM8S.h.

5.1.2.1466 _TIM5_TIE

```
#define _TIM5_TIE ((uint8_t) (0x01 << 6))
```

TIM5 Trigger interrupt enable [0] (in _TIM5_IER)

Definition at line 4264 of file STM8AF_STM8S.h.

5.1.2.1467 _TIM5_TIF

```
#define _TIM5_TIF ((uint8_t) (0x01 << 6))
```

TIM5 Trigger interrupt flag [0] (in _TIM5_SR1)

Definition at line 4273 of file STM8AF_STM8S.h.

5.1.2.1468 _TIM5_TS

```
#define _TIM5_TS ((uint8_t) (0x07 << 4))
```

TIM5 Trigger selection [2:0] (in _TIM5_SMCR)

Definition at line 4252 of file STM8AF_STM8S.h.

5.1.2.1469 _TIM5_TS0

```
#define _TIM5_TS0 ((uint8_t) (0x01 << 4))
```

TIM5 Trigger selection [0] (in _TIM5_SMCR)

Definition at line 4253 of file STM8AF_STM8S.h.

5.1.2.1470 _TIM5_TS1

```
#define _TIM5_TS1 ((uint8_t) (0x01 << 5))
```

TIM5 Trigger selection [1] (in _TIM5_SMCR)

Definition at line 4254 of file STM8AF_STM8S.h.

5.1.2.1471 _TIM5_TS2

```
#define _TIM5_TS2 ((uint8_t) (0x01 << 6))
```

TIM5 Trigger selection [2] (in _TIM5_SMCR)

Definition at line 4255 of file STM8AF_STM8S.h.

5.1.2.1472 _TIM5_UDIS

```
#define _TIM5_UDIS ((uint8_t) (0x01 << 1))
```

TIM5 Update disable [0] (in _TIM5_CR1)

Definition at line 4229 of file STM8AF_STM8S.h.

5.1.2.1473 _TIM5_UG

```
#define _TIM5_UG ((uint8_t) (0x01 << 0))
```

TIM5 Update generation [0] (in _TIM5_EGR)

Definition at line 4284 of file STM8AF_STM8S.h.

5.1.2.1474 _TIM5_UIE

```
#define _TIM5_UIE ((uint8_t) (0x01 << 0))
```

TIM5 Update interrupt enable [0] (in _TIM5_IER)

Definition at line 4259 of file STM8AF_STM8S.h.

5.1.2.1475 _TIM5_UIF

```
#define _TIM5_UIF ((uint8_t) (0x01 << 0))
```

TIM5 Update interrupt flag [0] (in _TIM5_SR1)

Definition at line 4268 of file STM8AF_STM8S.h.

5.1.2.1476 _TIM5_URS

```
#define _TIM5_URS ((uint8_t) (0x01 << 2))
```

TIM5 Update request source [0] (in _TIM5_CR1)

Definition at line 4230 of file STM8AF_STM8S.h.

5.1.2.1477 _TIM6

```
#define _TIM6 _SFR(TIM6_t, TIM6_AddressBase)
```

TIM6 struct/bit access.

Definition at line 4464 of file STM8AF_STM8S.h.

5.1.2.1478 _TIM6_ARPE

```
#define _TIM6_ARPE ((uint8_t) (0x01 << 7))
```

TIM6 Auto-reload preload enable [0] (in _TIM6_CR1)

Definition at line 4488 of file STM8AF_STM8S.h.

5.1.2.1479 _TIM6_ARR

```
#define _TIM6_ARR _SFR(uint8_t, TIM6_AddressBase+0x06)
```

TIM6 auto-reload register.

Definition at line 4471 of file STM8AF_STM8S.h.

5.1.2.1480 _TIM6_ARR_RESET_VALUE

```
#define _TIM6_ARR_RESET_VALUE ((uint8_t) 0xFF)
```

TIM6 auto-reload register reset value.

Definition at line 4480 of file STM8AF_STM8S.h.

5.1.2.1481 _TIM6_CEN

```
#define _TIM6_CEN ((uint8_t) (0x01 << 0))
```

TIM6 Counter enable [0] (in _TIM6_CR1)

Definition at line 4483 of file STM8AF_STM8S.h.

5.1.2.1482 _TIM6_CNTR

```
#define _TIM6_CNTR _SFR(uint8_t, TIM6_AddressBase+0x04)
```

TIM6 counter register.

Definition at line 4469 of file STM8AF_STM8S.h.

5.1.2.1483 _TIM6_CNTR_RESET_VALUE

```
#define _TIM6_CNTR_RESET_VALUE ((uint8_t) 0x00)
```

TIM6 counter register reset value.

Definition at line 4478 of file STM8AF_STM8S.h.

5.1.2.1484 _TIM6_CR

```
#define _TIM6_CR _SFR(uint8_t, TIM6_AddressBase+0x00)
```

TIM6 control register.

Definition at line 4465 of file STM8AF_STM8S.h.

5.1.2.1485 _TIM6_CR_RESET_VALUE

```
#define _TIM6_CR_RESET_VALUE ((uint8_t) 0x00)
```

TIM6 control register reset value.

Definition at line 4474 of file STM8AF_STM8S.h.

5.1.2.1486 _TIM6_EGR

```
#define _TIM6_EGR _SFR(uint8_t, TIM6_AddressBase+0x03)
```

TIM6 event generation register.

Definition at line 4468 of file STM8AF_STM8S.h.

5.1.2.1487 _TIM6_EGR_RESET_VALUE

```
#define _TIM6_EGR_RESET_VALUE ((uint8_t) 0x00)
```

TIM6 event generation register reset value.

Definition at line 4477 of file STM8AF_STM8S.h.

5.1.2.1488 _TIM6_IER

```
#define _TIM6_IER _SFR(uint8_t, TIM6_AddressBase+0x01)
```

TIM6 interrupt enable register.

Definition at line 4466 of file STM8AF_STM8S.h.

5.1.2.1489 _TIM6_IER_RESET_VALUE

```
#define _TIM6_IER_RESET_VALUE ((uint8_t) 0x00)
```

TIM6 interrupt enable register reset value.

Definition at line 4475 of file STM8AF_STM8S.h.

5.1.2.1490 _TIM6_MMS

```
#define _TIM6_MMS ((uint8_t) (0x07 << 4))
```

TIM6 Master mode selection [2:0] (in _TIM6_CR2)

Definition at line 4492 of file STM8AF_STM8S.h.

5.1.2.1491 _TIM6_MMS0

```
#define _TIM6_MMS0 ((uint8_t) (0x01 << 4))
```

TIM6 Master mode selection [0] (in _TIM6_CR2)

Definition at line 4493 of file STM8AF_STM8S.h.

5.1.2.1492 _TIM6_MMS1

```
#define _TIM6_MMS1 ((uint8_t) (0x01 << 5))
```

TIM6 Master mode selection [1] (in _TIM6_CR2)

Definition at line 4494 of file STM8AF_STM8S.h.

5.1.2.1493 _TIM6_MMS2

```
#define _TIM6_MMS2 ((uint8_t) (0x01 << 6))
```

TIM6 Master mode selection [2] (in _TIM6_CR2)

Definition at line 4495 of file STM8AF_STM8S.h.

5.1.2.1494 _TIM6_OPM

```
#define _TIM6_OPM ((uint8_t) (0x01 << 3))
```

TIM6 One-pulse mode [0] (in _TIM6_CR1)

Definition at line 4486 of file STM8AF_STM8S.h.

5.1.2.1495 _TIM6_PSC

```
#define _TIM6_PSC ((uint8_t) (0x07 << 0))
```

TIM6 clock prescaler [2:0] (in _TIM6_PSCR)

Definition at line 4523 of file STM8AF_STM8S.h.

5.1.2.1496 _TIM6_PSC0

```
#define _TIM6_PSC0 ((uint8_t) (0x01 << 0))
```

TIM6 clock prescaler [0] (in _TIM6_PSCR)

Definition at line 4524 of file STM8AF_STM8S.h.

5.1.2.1497 _TIM6_PSC1

```
#define _TIM6_PSC1 ((uint8_t) (0x01 << 1))
```

TIM6 clock prescaler [1] (in _TIM6_PSCR)

Definition at line 4525 of file STM8AF_STM8S.h.

5.1.2.1498 _TIM6_PSC2

```
#define _TIM6_PSC2 ((uint8_t) (0x01 << 2))
```

TIM6 clock prescaler [2] (in _TIM6_PSCR)

Definition at line 4526 of file STM8AF_STM8S.h.

5.1.2.1499 _TIM6_PSCR

```
#define _TIM6_PSCR _SFR(uint8_t, TIM6_AddressBase+0x05)
```

TIM6 clock prescaler register.

Definition at line 4470 of file STM8AF_STM8S.h.

5.1.2.1500 _TIM6_PSCR_RESET_VALUE

```
#define _TIM6_PSCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM6 clock prescaler register reset value.

Definition at line 4479 of file STM8AF_STM8S.h.

5.1.2.1501 _TIM6_SMS

```
#define _TIM6_SMS ((uint8_t) (0x07 << 0))
```

TIM6 Clock/trigger/slave mode selection [2:0] (in _TIM6_SMCR)

Definition at line 4499 of file STM8AF_STM8S.h.

5.1.2.1502 _TIM6_SMS0

```
#define _TIM6_SMS0 ((uint8_t) (0x01 << 0))
```

TIM6 Clock/trigger/slave mode selection [0] (in _TIM6_SMCR)

Definition at line 4500 of file STM8AF_STM8S.h.

5.1.2.1503 _TIM6_SMS1

```
#define _TIM6_SMS1 ((uint8_t) (0x01 << 1))
```

TIM6 Clock/trigger/slave mode selection [1] (in _TIM6_SMCR)

Definition at line 4501 of file STM8AF_STM8S.h.

5.1.2.1504 _TIM6_SMS2

```
#define _TIM6_SMS2 ((uint8_t) (0x01 << 2))
```

TIM6 Clock/trigger/slave mode selection [2] (in _TIM6_SMCR)

Definition at line 4502 of file STM8AF_STM8S.h.

5.1.2.1505 _TIM6_SR

```
#define _TIM6_SR _SFR(uint8_t, TIM6_AddressBase+0x02)
```

TIM6 status register.

Definition at line 4467 of file STM8AF_STM8S.h.

5.1.2.1506 _TIM6_SR_RESET_VALUE

```
#define _TIM6_SR_RESET_VALUE ((uint8_t) 0x00)
```

TIM6 status register reset value.

Definition at line 4476 of file STM8AF_STM8S.h.

5.1.2.1507 _TIM6_TS

```
#define _TIM6_TS ((uint8_t) (0x07 << 4))
```

TIM6 Trigger selection [2:0] (in _TIM6_SMCR)

Definition at line 4504 of file STM8AF_STM8S.h.

5.1.2.1508 _TIM6_TS0

```
#define _TIM6_TS0 ((uint8_t) (0x01 << 4))
```

TIM6 Trigger selection [0] (in _TIM6_SMCR)

Definition at line 4505 of file STM8AF_STM8S.h.

5.1.2.1509 _TIM6_TS1

```
#define _TIM6_TS1 ((uint8_t) (0x01 << 5))
```

TIM6 Trigger selection [1] (in _TIM6_SMCR)

Definition at line 4506 of file STM8AF_STM8S.h.

5.1.2.1510 _TIM6_TS2

```
#define _TIM6_TS2 ((uint8_t) (0x01 << 6))
```

TIM6 Trigger selection [2] (in _TIM6_SMCR)

Definition at line 4507 of file STM8AF_STM8S.h.

5.1.2.1511 _TIM6_UDIS

```
#define _TIM6_UDIS ((uint8_t) (0x01 << 1))
```

TIM6 Update disable [0] (in _TIM6_CR1)

Definition at line 4484 of file STM8AF_STM8S.h.

5.1.2.1512 _TIM6_UG

```
#define _TIM6_UG ((uint8_t) (0x01 << 0))
```

TIM6 Update generation [0] (in _TIM6_EGR)

Definition at line 4519 of file STM8AF_STM8S.h.

5.1.2.1513 _TIM6_UIE

```
#define _TIM6_UIE ((uint8_t) (0x01 << 0))
```

TIM6 Update interrupt enable [0] (in _TIM6_IER)

Definition at line 4511 of file STM8AF_STM8S.h.

5.1.2.1514 _TIM6_UIF

```
#define _TIM6_UIF ((uint8_t) (0x01 << 0))
```

TIM6 Update interrupt flag [0] (in _TIM6_SR)

Definition at line 4515 of file STM8AF_STM8S.h.

5.1.2.1515 _TIM6_URS

```
#define _TIM6_URS ((uint8_t) (0x01 << 2))
```

TIM6 Update request source [0] (in _TIM6_CR1)

Definition at line 4485 of file STM8AF_STM8S.h.

5.1.2.1516 _UART1

```
#define _UART1 _SFR(UART1_t, UART1_AddressBase)
```

UART1 struct/bit access.

Definition at line 1731 of file STM8AF_STM8S.h.

5.1.2.1517 _UART1_ADD

```
#define _UART1_ADD ((uint8_t) (0x0F << 0))
```

UART1 Address of the UART node [3:0] (in _UART1_CR4)

Definition at line 1798 of file STM8AF_STM8S.h.

5.1.2.1518 _UART1_ADD0

```
#define _UART1_ADD0 ((uint8_t) (0x01 << 0))
```

UART1 Address of the UART node [0] (in _UART1_CR4)

Definition at line 1799 of file STM8AF_STM8S.h.

5.1.2.1519 _UART1_ADD1

```
#define _UART1_ADD1 ((uint8_t) (0x01 << 1))
```

UART1 Address of the UART node [1] (in _UART1_CR4)

Definition at line 1800 of file STM8AF_STM8S.h.

5.1.2.1520 _UART1_ADD2

```
#define _UART1_ADD2 ((uint8_t) (0x01 << 2))
```

UART1 Address of the UART node [2] (in _UART1_CR4)

Definition at line 1801 of file STM8AF_STM8S.h.

5.1.2.1521 _UART1_ADD3

```
#define _UART1_ADD3 ((uint8_t) (0x01 << 3))
```

UART1 Address of the UART node [3] (in _UART1_CR4)

Definition at line 1802 of file STM8AF_STM8S.h.

5.1.2.1522 _UART1_BRR1

```
#define _UART1_BRR1 _SFR(uint8_t, UART1_AddressBase+0x02)
```

UART1 Baud rate register 1.

Definition at line 1734 of file STM8AF_STM8S.h.

5.1.2.1523 _UART1_BRR1_RESET_VALUE

```
#define _UART1_BRR1_RESET_VALUE ((uint8_t) 0x00)
```

UART1 Baud rate register 1 reset value.

Definition at line 1746 of file STM8AF_STM8S.h.

5.1.2.1524 _UART1_BRR2

```
#define _UART1_BRR2 _SFR(uint8_t, UART1_AddressBase+0x03)
```

UART1 Baud rate register 2.

Definition at line 1735 of file STM8AF_STM8S.h.

5.1.2.1525 _UART1_BRR2_RESET_VALUE

```
#define _UART1_BRR2_RESET_VALUE ((uint8_t) 0x00)
```

UART1 Baud rate register 2 reset value.

Definition at line 1747 of file STM8AF_STM8S.h.

5.1.2.1526 _UART1_CKEN

```
#define _UART1_CKEN ((uint8_t) (0x01 << 3))
```

UART1 Clock enable [0] (in _UART1_CR3)

Definition at line 1790 of file STM8AF_STM8S.h.

5.1.2.1527 _UART1_CPHA

```
#define _UART1_CPHA ((uint8_t) (0x01 << 1))
```

UART1 Clock phase [0] (in _UART1_CR3)

Definition at line 1788 of file STM8AF_STM8S.h.

5.1.2.1528 _UART1_CPOL

```
#define _UART1_CPOL ((uint8_t) (0x01 << 2))
```

UART1 Clock polarity [0] (in _UART1_CR3)

Definition at line 1789 of file STM8AF_STM8S.h.

5.1.2.1529 _UART1_CR1

```
#define _UART1_CR1 _SFR(uint8_t, UART1_AddressBase+0x04)
```

UART1 Control register 1.

Definition at line 1736 of file STM8AF_STM8S.h.

5.1.2.1530 _UART1_CR1_RESET_VALUE

```
#define _UART1_CR1_RESET_VALUE ((uint8_t) 0x00)
```

UART1 Control register 1 reset value.

Definition at line 1748 of file STM8AF_STM8S.h.

5.1.2.1531 _UART1_CR2

```
#define _UART1_CR2 _SFR(uint8_t, UART1_AddressBase+0x05)
```

UART1 Control register 2.

Definition at line 1737 of file STM8AF_STM8S.h.

5.1.2.1532 _UART1_CR2_RESET_VALUE

```
#define _UART1_CR2_RESET_VALUE ((uint8_t) 0x00)
```

UART1 Control register 2 reset value.

Definition at line 1749 of file STM8AF_STM8S.h.

5.1.2.1533 _UART1_CR3

```
#define _UART1_CR3 _SFR(uint8_t, UART1_AddressBase+0x06)
```

UART1 Control register 3.

Definition at line 1738 of file STM8AF_STM8S.h.

5.1.2.1534 _UART1_CR3_RESET_VALUE

```
#define _UART1_CR3_RESET_VALUE ((uint8_t) 0x00)
```

UART1 Control register 3 reset value.

Definition at line 1750 of file STM8AF_STM8S.h.

5.1.2.1535 _UART1_CR4

```
#define _UART1_CR4 _SFR(uint8_t, UART1_AddressBase+0x07)
```

UART1 Control register 4.

Definition at line 1739 of file STM8AF_STM8S.h.

5.1.2.1536 _UART1_CR4_RESET_VALUE

```
#define _UART1_CR4_RESET_VALUE ((uint8_t) 0x00)
```

UART1 Control register 4 reset value.

Definition at line 1751 of file STM8AF_STM8S.h.

5.1.2.1537 _UART1_CR5

```
#define _UART1_CR5 _SFR(uint8_t, UART1_AddressBase+0x08)
```

UART1 Control register 5.

Definition at line 1740 of file STM8AF_STM8S.h.

5.1.2.1538 _UART1_CR5_RESET_VALUE

```
#define _UART1_CR5_RESET_VALUE ((uint8_t) 0x00)
```

UART1 Control register 5 reset value.

Definition at line 1752 of file STM8AF_STM8S.h.

5.1.2.1539 _UART1_DR

```
#define _UART1_DR _SFR(uint8_t, UART1_AddressBase+0x01)
```

UART1 data register.

Definition at line 1733 of file STM8AF_STM8S.h.

5.1.2.1540 _UART1_FE

```
#define _UART1_FE ((uint8_t) (0x01 << 1))
```

UART1 Framing error [0] (in _UART1_SR)

Definition at line 1758 of file STM8AF_STM8S.h.

5.1.2.1541 _UART1_GTR

```
#define _UART1_GTR _SFR(uint8_t, UART1_AddressBase+0x09)
```

UART1 guard time register.

Definition at line 1741 of file STM8AF_STM8S.h.

5.1.2.1542 _UART1_GTR_RESET_VALUE

```
#define _UART1_GTR_RESET_VALUE ((uint8_t) 0x00)
```

UART1 guard time register reset value.

Definition at line 1753 of file STM8AF_STM8S.h.

5.1.2.1543 _UART1_HDSEL

```
#define _UART1_HDSEL ((uint8_t) (0x01 << 3))
```

UART1 Half-Duplex Selection [0] (in _UART1_CR5)

Definition at line 1812 of file STM8AF_STM8S.h.

5.1.2.1544 _UART1_IDLE

```
#define _UART1_IDLE ((uint8_t) (0x01 << 4))
```

UART1 IDLE line detected [0] (in _UART1_SR)

Definition at line 1761 of file STM8AF_STM8S.h.

5.1.2.1545 _UART1_ILIEN

```
#define _UART1_ILIEN ((uint8_t) (0x01 << 4))
```

UART1 IDLE Line interrupt enable [0] (in _UART1_CR2)

Definition at line 1781 of file STM8AF_STM8S.h.

5.1.2.1546 _UART1_IREN

```
#define _UART1_IREN ((uint8_t) (0x01 << 1))
```

UART1 IrDA mode Enable [0] (in _UART1_CR5)

Definition at line 1810 of file STM8AF_STM8S.h.

5.1.2.1547 _UART1_IRLP

```
#define _UART1_IRLP ((uint8_t) (0x01 << 2))
```

UART1 IrDA Low Power [0] (in _UART1_CR5)

Definition at line 1811 of file STM8AF_STM8S.h.

5.1.2.1548 _UART1_LBCL

```
#define _UART1_LBCL ((uint8_t) (0x01 << 0))
```

UART1 Last bit clock pulse [0] (in _UART1_CR3)

Definition at line 1787 of file STM8AF_STM8S.h.

5.1.2.1549 _UART1_LBDF

```
#define _UART1_LBDF ((uint8_t) (0x01 << 4))
```

UART1 LIN Break Detection Flag [0] (in _UART1_CR4)

Definition at line 1803 of file STM8AF_STM8S.h.

5.1.2.1550 _UART1_LBDIEN

```
#define _UART1_LBDIEN ((uint8_t) (0x01 << 6))
```

UART1 LIN Break Detection Interrupt Enable [0] (in _UART1_CR4)

Definition at line 1805 of file STM8AF_STM8S.h.

5.1.2.1551 _UART1_LBDL

```
#define _UART1_LBDL ((uint8_t) (0x01 << 5))
```

UART1 LIN Break Detection Length [0] (in _UART1_CR4)

Definition at line 1804 of file STM8AF_STM8S.h.

5.1.2.1552 _UART1_LINEN

```
#define _UART1_LINEN ((uint8_t) (0x01 << 6))
```

UART1 LIN mode enable [0] (in _UART1_CR3)

Definition at line 1794 of file STM8AF_STM8S.h.

5.1.2.1553 _UART1_M

```
#define _UART1_M ((uint8_t) (0x01 << 4))
```

UART1 word length [0] (in _UART1_CR1)

Definition at line 1771 of file STM8AF_STM8S.h.

5.1.2.1554 _UART1_NACK

```
#define _UART1_NACK ((uint8_t) (0x01 << 4))
```

UART1 Smartcard NACK enable [0] (in _UART1_CR5)

Definition at line 1813 of file STM8AF_STM8S.h.

5.1.2.1555 _UART1_NF

```
#define _UART1_NF ((uint8_t) (0x01 << 2))
```

UART1 Noise flag [0] (in _UART1_SR)

Definition at line 1759 of file STM8AF_STM8S.h.

5.1.2.1556 _UART1_OR_LHE

```
#define _UART1_OR_LHE ((uint8_t) (0x01 << 3))
```

UART1 LIN Header Error (LIN slave mode) / Overrun error [0] (in _UART1_SR)

Definition at line 1760 of file STM8AF_STM8S.h.

5.1.2.1557 _UART1_PCEN

```
#define _UART1_PCEN ((uint8_t) (0x01 << 2))
```

UART1 Parity control enable [0] (in _UART1_CR1)

Definition at line 1769 of file STM8AF_STM8S.h.

5.1.2.1558 _UART1_PE

```
#define _UART1_PE ((uint8_t) (0x01 << 0))
```

UART1 Parity error [0] (in _UART1_SR)

Definition at line 1757 of file STM8AF_STM8S.h.

5.1.2.1559 _UART1_PIEN

```
#define _UART1_PIEN ((uint8_t) (0x01 << 0))
```

UART1 Parity interrupt enable [0] (in _UART1_CR1)

Definition at line 1767 of file STM8AF_STM8S.h.

5.1.2.1560 _UART1_PS

```
#define _UART1_PS ((uint8_t) (0x01 << 1))
```

UART1 Parity selection [0] (in _UART1_CR1)

Definition at line 1768 of file STM8AF_STM8S.h.

5.1.2.1561 _UART1_PSCR

```
#define _UART1_PSCR _SFR(uint8_t, UART1_AddressBase+0x0A)
```

UART1 prescaler register.

Definition at line 1742 of file STM8AF_STM8S.h.

5.1.2.1562 _UART1_PSCR_RESET_VALUE

```
#define _UART1_PSCR_RESET_VALUE ((uint8_t) 0x00)
```

UART1 prescaler register reset value.

Definition at line 1754 of file STM8AF_STM8S.h.

5.1.2.1563 _UART1_R8

```
#define _UART1_R8 ((uint8_t) (0x01 << 7))
```

UART1 Receive Data bit 8 (in 9-bit mode) [0] (in _UART1_CR1)

Definition at line 1774 of file STM8AF_STM8S.h.

5.1.2.1564 _UART1_REN

```
#define _UART1_REN ((uint8_t) (0x01 << 2))
```

UART1 Receiver enable [0] (in _UART1_CR2)

Definition at line 1779 of file STM8AF_STM8S.h.

5.1.2.1565 _UART1_RIEN

```
#define _UART1_RIEN ((uint8_t) (0x01 << 5))
```

UART1 Receiver interrupt enable [0] (in _UART1_CR2)

Definition at line 1782 of file STM8AF_STM8S.h.

5.1.2.1566 _UART1_RWU

```
#define _UART1_RWU ((uint8_t) (0x01 << 1))
```

UART1 Receiver wakeup [0] (in _UART1_CR2)

Definition at line 1778 of file STM8AF_STM8S.h.

5.1.2.1567 _UART1_RXNE

```
#define _UART1_RXNE ((uint8_t) (0x01 << 5))
```

UART1 Read data register not empty [0] (in _UART1_SR)

Definition at line 1762 of file STM8AF_STM8S.h.

5.1.2.1568 _UART1_SBK

```
#define _UART1_SBK ((uint8_t) (0x01 << 0))
```

UART1 Send break [0] (in _UART1_CR2)

Definition at line 1777 of file STM8AF_STM8S.h.

5.1.2.1569 _UART1_SCEN

```
#define _UART1_SCEN ((uint8_t) (0x01 << 5))
```

UART1 Smartcard mode enable [0] (in _UART1_CR5)

Definition at line 1814 of file STM8AF_STM8S.h.

5.1.2.1570 _UART1_SR

```
#define _UART1_SR _SFR(uint8_t, UART1_AddressBase+0x00)
```

UART1 Status register.

Definition at line 1732 of file STM8AF_STM8S.h.

5.1.2.1571 _UART1_SR_RESET_VALUE

```
#define _UART1_SR_RESET_VALUE ((uint8_t) 0xC0)
```

UART1 Status register reset value.

Definition at line 1745 of file STM8AF_STM8S.h.

5.1.2.1572 _UART1_STOP

```
#define _UART1_STOP ((uint8_t) (0x03 << 4))
```

UART1 STOP bits [1:0] (in _UART1_CR3)

Definition at line 1791 of file STM8AF_STM8S.h.

5.1.2.1573 _UART1_STOP0

```
#define _UART1_STOP0 ((uint8_t) (0x01 << 4))
```

UART1 STOP bits [0] (in _UART1_CR3)

Definition at line 1792 of file STM8AF_STM8S.h.

5.1.2.1574 _UART1_STOP1

```
#define _UART1_STOP1 ((uint8_t) (0x01 << 5))
```

UART1 STOP bits [1] (in _UART1_CR3)

Definition at line 1793 of file STM8AF_STM8S.h.

5.1.2.1575 _UART1_T8

```
#define _UART1_T8 ((uint8_t) (0x01 << 6))
```

UART1 Transmit Data bit 8 (in 9-bit mode) [0] (in _UART1_CR1)

Definition at line 1773 of file STM8AF_STM8S.h.

5.1.2.1576 _UART1_TC

```
#define _UART1_TC ((uint8_t) (0x01 << 6))
```

UART1 Transmission complete [0] (in _UART1_SR)

Definition at line 1763 of file STM8AF_STM8S.h.

5.1.2.1577 _UART1_TCIEN

```
#define _UART1_TCIEN ((uint8_t) (0x01 << 6))
```

UART1 Transmission complete interrupt enable [0] (in _UART1_CR2)

Definition at line 1783 of file STM8AF_STM8S.h.

5.1.2.1578 _UART1_TEN

```
#define _UART1_TEN ((uint8_t) (0x01 << 3))
```

UART1 Transmitter enable [0] (in _UART1_CR2)

Definition at line 1780 of file STM8AF_STM8S.h.

5.1.2.1579 _UART1_TIEN

```
#define _UART1_TIEN ((uint8_t) (0x01 << 7))
```

UART1 Transmitter interrupt enable [0] (in _UART1_CR2)

Definition at line 1784 of file STM8AF_STM8S.h.

5.1.2.1580 _UART1_TXE

```
#define _UART1_TXE ((uint8_t) (0x01 << 7))
```

UART1 Transmit data register empty [0] (in _UART1_SR)

Definition at line 1764 of file STM8AF_STM8S.h.

5.1.2.1581 _UART1_UARTD

```
#define _UART1_UARTD ((uint8_t) (0x01 << 5))
```

UART1 Disable (for low power consumption) [0] (in _UART1_CR1)

Definition at line 1772 of file STM8AF_STM8S.h.

5.1.2.1582 _UART1_WAKE

```
#define _UART1_WAKE ((uint8_t) (0x01 << 3))
```

UART1 Wakeup method [0] (in _UART1_CR1)

Definition at line 1770 of file STM8AF_STM8S.h.

5.1.2.1583 _UART2

```
#define _UART2 _SFR(UART2_t, UART2_AddressBase)
```

UART2 struct/bit access.

Definition at line 1948 of file STM8AF_STM8S.h.

5.1.2.1584 _UART2_ADD

```
#define _UART2_ADD ((uint8_t) (0x0F << 0))
```

UART2 Address of the UART node [3:0] (in _UART2_CR4)

Definition at line 2017 of file STM8AF_STM8S.h.

5.1.2.1585 _UART2_ADD0

```
#define _UART2_ADD0 ((uint8_t) (0x01 << 0))
```

UART2 Address of the UART node [0] (in _UART2_CR4)

Definition at line 2018 of file STM8AF_STM8S.h.

5.1.2.1586 _UART2_ADD1

```
#define _UART2_ADD1 ((uint8_t) (0x01 << 1))
```

UART2 Address of the UART node [1] (in _UART2_CR4)

Definition at line 2019 of file STM8AF_STM8S.h.

5.1.2.1587 _UART2_ADD2

```
#define _UART2_ADD2 ((uint8_t) (0x01 << 2))
```

UART2 Address of the UART node [2] (in _UART2_CR4)

Definition at line 2020 of file STM8AF_STM8S.h.

5.1.2.1588 _UART2_ADD3

```
#define _UART2_ADD3 ((uint8_t) (0x01 << 3))
```

UART2 Address of the UART node [3] (in _UART2_CR4)

Definition at line 2021 of file STM8AF_STM8S.h.

5.1.2.1589 _UART2_BRR1

```
#define _UART2_BRR1 _SFR(uint8_t, UART2_AddressBase+0x02)
```

UART2 Baud rate register 1.

Definition at line 1951 of file STM8AF_STM8S.h.

5.1.2.1590 _UART2_BRR1_RESET_VALUE

```
#define _UART2_BRR1_RESET_VALUE ((uint8_t) 0x00)
```

UART2 Baud rate register 1 reset value.

Definition at line 1964 of file STM8AF_STM8S.h.

5.1.2.1591 _UART2_BRR2

```
#define _UART2_BRR2 _SFR(uint8_t, UART2_AddressBase+0x03)
```

UART2 Baud rate register 2.

Definition at line 1952 of file STM8AF_STM8S.h.

5.1.2.1592 _UART2_BRR2_RESET_VALUE

```
#define _UART2_BRR2_RESET_VALUE ((uint8_t) 0x00)
```

UART2 Baud rate register 2 reset value.

Definition at line 1965 of file STM8AF_STM8S.h.

5.1.2.1593 _UART2_CKEN

```
#define _UART2_CKEN ((uint8_t) (0x01 << 3))
```

UART2 Clock enable [0] (in _UART2_CR3)

Definition at line 2009 of file STM8AF_STM8S.h.

5.1.2.1594 _UART2_CPHA

```
#define _UART2_CPHA ((uint8_t) (0x01 << 1))
```

UART2 Clock phase [0] (in _UART2_CR3)

Definition at line 2007 of file STM8AF_STM8S.h.

5.1.2.1595 _UART2_CPOL

```
#define _UART2_CPOL ((uint8_t) (0x01 << 2))
```

UART2 Clock polarity [0] (in _UART2_CR3)

Definition at line 2008 of file STM8AF_STM8S.h.

5.1.2.1596 _UART2_CR1

```
#define _UART2_CR1 _SFR(uint8_t, UART2_AddressBase+0x04)
```

UART2 Control register 1.

Definition at line 1953 of file STM8AF_STM8S.h.

5.1.2.1597 _UART2_CR1_RESET_VALUE

```
#define _UART2_CR1_RESET_VALUE ((uint8_t) 0x00)
```

UART2 Control register 1 reset value.

Definition at line 1966 of file STM8AF_STM8S.h.

5.1.2.1598 _UART2_CR2

```
#define _UART2_CR2 _SFR(uint8_t, UART2_AddressBase+0x05)
```

UART2 Control register 2.

Definition at line 1954 of file STM8AF_STM8S.h.

5.1.2.1599 _UART2_CR2_RESET_VALUE

```
#define _UART2_CR2_RESET_VALUE ((uint8_t) 0x00)
```

UART2 Control register 2 reset value.

Definition at line 1967 of file STM8AF_STM8S.h.

5.1.2.1600 _UART2_CR3

```
#define _UART2_CR3 _SFR(uint8_t, UART2_AddressBase+0x06)
```

UART2 Control register 3.

Definition at line 1955 of file STM8AF_STM8S.h.

5.1.2.1601 _UART2_CR3_RESET_VALUE

```
#define _UART2_CR3_RESET_VALUE ((uint8_t) 0x00)
```

UART2 Control register 3 reset value.

Definition at line 1968 of file STM8AF_STM8S.h.

5.1.2.1602 _UART2_CR4

```
#define _UART2_CR4 _SFR(uint8_t, UART2_AddressBase+0x07)
```

UART2 Control register 4.

Definition at line 1956 of file STM8AF_STM8S.h.

5.1.2.1603 _UART2_CR4_RESET_VALUE

```
#define _UART2_CR4_RESET_VALUE ((uint8_t) 0x00)
```

UART2 Control register 4 reset value.

Definition at line 1969 of file STM8AF_STM8S.h.

5.1.2.1604 _UART2_CR5

```
#define _UART2_CR5 _SFR(uint8_t, UART2_AddressBase+0x08)
```

UART2 Control register 5.

Definition at line 1957 of file STM8AF_STM8S.h.

5.1.2.1605 _UART2_CR5_RESET_VALUE

```
#define _UART2_CR5_RESET_VALUE ((uint8_t) 0x00)
```

UART2 Control register 5 reset value.

Definition at line 1970 of file STM8AF_STM8S.h.

5.1.2.1606 _UART2_CR6

```
#define _UART2_CR6 _SFR(uint8_t, UART2_AddressBase+0x09)
```

UART2 Control register 6.

Definition at line 1958 of file STM8AF_STM8S.h.

5.1.2.1607 _UART2_CR6_RESET_VALUE

```
#define _UART2_CR6_RESET_VALUE ((uint8_t) 0x00)
```

UART2 Control register 6 reset value.

Definition at line 1971 of file STM8AF_STM8S.h.

5.1.2.1608 _UART2_DR

```
#define _UART2_DR _SFR(uint8_t, UART2_AddressBase+0x01)
```

UART2 data register.

Definition at line 1950 of file STM8AF_STM8S.h.

5.1.2.1609 _UART2_FE

```
#define _UART2_FE ((uint8_t) (0x01 << 1))
```

UART2 Framing error [0] (in _UART2_SR)

Definition at line 1977 of file STM8AF_STM8S.h.

5.1.2.1610 _UART2_GTR

```
#define _UART2_GTR _SFR(uint8_t, UART2_AddressBase+0x0A)
```

UART2 guard time register.

Definition at line 1959 of file STM8AF_STM8S.h.

5.1.2.1611 _UART2_GTR_RESET_VALUE

```
#define _UART2_GTR_RESET_VALUE ((uint8_t) 0x00)
```

UART2 guard time register reset value.

Definition at line 1972 of file STM8AF_STM8S.h.

5.1.2.1612 _UART2_IDLE

```
#define _UART2_IDLE ((uint8_t) (0x01 << 4))
```

UART2 IDLE line detected [0] (in _UART2_SR)

Definition at line 1980 of file STM8AF_STM8S.h.

5.1.2.1613 _UART2_ILIEN

```
#define _UART2_ILIEN ((uint8_t) (0x01 << 4))
```

UART2 IDLE Line interrupt enable [0] (in _UART2_CR2)

Definition at line 2000 of file STM8AF_STM8S.h.

5.1.2.1614 _UART2_IREN

```
#define _UART2_IREN ((uint8_t) (0x01 << 1))
```

UART2 IrDA mode Enable [0] (in _UART2_CR5)

Definition at line 2029 of file STM8AF_STM8S.h.

5.1.2.1615 _UART2_IRLP

```
#define _UART2_IRLP ((uint8_t) (0x01 << 2))
```

UART2 IrDA Low Power [0] (in _UART2_CR5)

Definition at line 2030 of file STM8AF_STM8S.h.

5.1.2.1616 _UART2_LASE

```
#define _UART2_LASE ((uint8_t) (0x01 << 4))
```

UART2 LIN automatic resynchronisation enable [0] (in _UART2_CR6)

Definition at line 2041 of file STM8AF_STM8S.h.

5.1.2.1617 _UART2_LBCL

```
#define _UART2_LBCL ((uint8_t) (0x01 << 0))
```

UART2 Last bit clock pulse [0] (in _UART2_CR3)

Definition at line 2006 of file STM8AF_STM8S.h.

5.1.2.1618 _UART2_LBDF

```
#define _UART2_LBDF ((uint8_t) (0x01 << 4))
```

UART2 LIN Break Detection Flag [0] (in _UART2_CR4)

Definition at line 2022 of file STM8AF_STM8S.h.

5.1.2.1619 _UART2_LBDIEN

```
#define _UART2_LBDIEN ((uint8_t) (0x01 << 6))
```

UART2 LIN Break Detection Interrupt Enable [0] (in _UART2_CR4)

Definition at line 2024 of file STM8AF_STM8S.h.

5.1.2.1620 _UART2_LBDL

```
#define _UART2_LBDL ((uint8_t) (0x01 << 5))
```

UART2 LIN Break Detection Length [0] (in _UART2_CR4)

Definition at line 2023 of file STM8AF_STM8S.h.

5.1.2.1621 _UART2_LDUM

```
#define _UART2_LDUM ((uint8_t) (0x01 << 7))
```

UART2 LIN Divider Update Method [0] (in _UART2_CR6)

Definition at line 2044 of file STM8AF_STM8S.h.

5.1.2.1622 _UART2_LHDF

```
#define _UART2_LHDF ((uint8_t) (0x01 << 1))
```

UART2 LIN Header Detection Flag [0] (in _UART2_CR6)

Definition at line 2038 of file STM8AF_STM8S.h.

5.1.2.1623 _UART2_LHDIEN

```
#define _UART2_LHDIEN ((uint8_t) (0x01 << 2))
```

UART2 LIN Header Detection Interrupt Enable [0] (in _UART2_CR6)

Definition at line 2039 of file STM8AF_STM8S.h.

5.1.2.1624 _UART2_LINEN

```
#define _UART2_LINEN ((uint8_t) (0x01 << 6))
```

UART2 LIN mode enable [0] (in _UART2_CR3)

Definition at line 2013 of file STM8AF_STM8S.h.

5.1.2.1625 _UART2_LSF

```
#define _UART2_LSF ((uint8_t) (0x01 << 0))
```

UART2 LIN Sync Field [0] (in _UART2_CR6)

Definition at line 2037 of file STM8AF_STM8S.h.

5.1.2.1626 _UART2_LSLV

```
#define _UART2_LSLV ((uint8_t) (0x01 << 5))
```

UART2 LIN Slave Enable [0] (in _UART2_CR6)

Definition at line 2042 of file STM8AF_STM8S.h.

5.1.2.1627 _UART2_M

```
#define _UART2_M ((uint8_t) (0x01 << 4))
```

UART2 word length [0] (in _UART2_CR1)

Definition at line 1990 of file STM8AF_STM8S.h.

5.1.2.1628 _UART2_NACK

```
#define _UART2_NACK ((uint8_t) (0x01 << 4))
```

UART2 Smartcard NACK enable [0] (in _UART2_CR5)

Definition at line 2032 of file STM8AF_STM8S.h.

5.1.2.1629 _UART2_NF

```
#define _UART2_NF ((uint8_t) (0x01 << 2))
```

UART2 Noise flag [0] (in _UART2_SR)

Definition at line 1978 of file STM8AF_STM8S.h.

5.1.2.1630 _UART2_OR_LHE

```
#define _UART2_OR_LHE ((uint8_t) (0x01 << 3))
```

UART2 LIN Header Error (LIN slave mode) / Overrun error [0] (in _UART2_SR)

Definition at line 1979 of file STM8AF_STM8S.h.

5.1.2.1631 _UART2_PCEN

```
#define _UART2_PCEN ((uint8_t) (0x01 << 2))
```

UART2 Parity control enable [0] (in _UART2_CR1)

Definition at line 1988 of file STM8AF_STM8S.h.

5.1.2.1632 _UART2_PE

```
#define _UART2_PE ((uint8_t) (0x01 << 0))
```

UART2 Parity error [0] (in _UART2_SR)

Definition at line 1976 of file STM8AF_STM8S.h.

5.1.2.1633 _UART2_PIEN

```
#define _UART2_PIEN ((uint8_t) (0x01 << 0))
```

UART2 Parity interrupt enable [0] (in _UART2_CR1)

Definition at line 1986 of file STM8AF_STM8S.h.

5.1.2.1634 _UART2_PS

```
#define _UART2_PS ((uint8_t) (0x01 << 1))
```

UART2 Parity selection [0] (in _UART2_CR1)

Definition at line 1987 of file STM8AF_STM8S.h.

5.1.2.1635 _UART2_PSCR

```
#define _UART2_PSCR _SFR(uint8_t, UART2_AddressBase+0x0B)
```

UART2 prescaler register.

Definition at line 1960 of file STM8AF_STM8S.h.

5.1.2.1636 _UART2_PSCR_RESET_VALUE

```
#define _UART2_PSCR_RESET_VALUE ((uint8_t) 0x00)
```

UART2 prescaler register reset value.

Definition at line 1973 of file STM8AF_STM8S.h.

5.1.2.1637 _UART2_R8

```
#define _UART2_R8 ((uint8_t) (0x01 << 7))
```

UART2 Receive Data bit 8 (in 9-bit mode) [0] (in _UART2_CR1)

Definition at line 1993 of file STM8AF_STM8S.h.

5.1.2.1638 _UART2_REN

```
#define _UART2_REN ((uint8_t) (0x01 << 2))
```

UART2 Receiver enable [0] (in _UART2_CR2)

Definition at line 1998 of file STM8AF_STM8S.h.

5.1.2.1639 _UART2_RIEN

```
#define _UART2_RIEN ((uint8_t) (0x01 << 5))
```

UART2 Receiver interrupt enable [0] (in _UART2_CR2)

Definition at line 2001 of file STM8AF_STM8S.h.

5.1.2.1640 _UART2_RWU

```
#define _UART2_RWU ((uint8_t) (0x01 << 1))
```

UART2 Receiver wakeup [0] (in _UART2_CR2)

Definition at line 1997 of file STM8AF_STM8S.h.

5.1.2.1641 _UART2_RXNE

```
#define _UART2_RXNE ((uint8_t) (0x01 << 5))
```

UART2 Read data register not empty [0] (in _UART2_SR)

Definition at line 1981 of file STM8AF_STM8S.h.

5.1.2.1642 _UART2_SBK

```
#define _UART2_SBK ((uint8_t) (0x01 << 0))
```

UART2 Send break [0] (in _UART2_CR2)

Definition at line 1996 of file STM8AF_STM8S.h.

5.1.2.1643 _UART2_SCEN

```
#define _UART2_SCEN ((uint8_t) (0x01 << 5))
```

UART2 Smartcard mode enable [0] (in _UART2_CR5)

Definition at line 2033 of file STM8AF_STM8S.h.

5.1.2.1644 _UART2_SR

```
#define _UART2_SR _SFR(uint8_t, UART2_AddressBase+0x00)
```

UART2 Status register.

Definition at line 1949 of file STM8AF_STM8S.h.

5.1.2.1645 _UART2_SR_RESET_VALUE

```
#define _UART2_SR_RESET_VALUE ((uint8_t) 0xC0)
```

UART2 Status register reset value.

Definition at line 1963 of file STM8AF_STM8S.h.

5.1.2.1646 _UART2_STOP

```
#define _UART2_STOP ((uint8_t) (0x03 << 4))
```

UART2 STOP bits [1:0] (in _UART2_CR3)

Definition at line 2010 of file STM8AF_STM8S.h.

5.1.2.1647 _UART2_STOP0

```
#define _UART2_STOP0 ((uint8_t) (0x01 << 4))
```

UART2 STOP bits [0] (in _UART2_CR3)

Definition at line 2011 of file STM8AF_STM8S.h.

5.1.2.1648 _UART2_STOP1

```
#define _UART2_STOP1 ((uint8_t) (0x01 << 5))
```

UART2 STOP bits [1] (in _UART2_CR3)

Definition at line 2012 of file STM8AF_STM8S.h.

5.1.2.1649 _UART2_T8

```
#define _UART2_T8 ((uint8_t) (0x01 << 6))
```

UART2 Transmit Data bit 8 (in 9-bit mode) [0] (in _UART2_CR1)

Definition at line 1992 of file STM8AF_STM8S.h.

5.1.2.1650 _UART2_TC

```
#define _UART2_TC ((uint8_t) (0x01 << 6))
```

UART2 Transmission complete [0] (in _UART2_SR)

Definition at line 1982 of file STM8AF_STM8S.h.

5.1.2.1651 _UART2_TCIEN

```
#define _UART2_TCIEN ((uint8_t) (0x01 << 6))
```

UART2 Transmission complete interrupt enable [0] (in _UART2_CR2)

Definition at line 2002 of file STM8AF_STM8S.h.

5.1.2.1652 _UART2_TEN

```
#define _UART2_TEN ((uint8_t) (0x01 << 3))
```

UART2 Transmitter enable [0] (in _UART2_CR2)

Definition at line 1999 of file STM8AF_STM8S.h.

5.1.2.1653 _UART2_TIEN

```
#define _UART2_TIEN ((uint8_t) (0x01 << 7))
```

UART2 Transmitter interrupt enable [0] (in _UART2_CR2)

Definition at line 2003 of file STM8AF_STM8S.h.

5.1.2.1654 _UART2_TXE

```
#define _UART2_TXE ((uint8_t) (0x01 << 7))
```

UART2 Transmit data register empty [0] (in _UART2_SR)

Definition at line 1983 of file STM8AF_STM8S.h.

5.1.2.1655 _UART2_UARTD

```
#define _UART2_UARTD ((uint8_t) (0x01 << 5))
```

UART2 Disable (for low power consumption) [0] (in _UART2_CR1)

Definition at line 1991 of file STM8AF_STM8S.h.

5.1.2.1656 _UART2_WAKE

```
#define _UART2_WAKE ((uint8_t) (0x01 << 3))
```

UART2 Wakeup method [0] (in _UART2_CR1)

Definition at line 1989 of file STM8AF_STM8S.h.

5.1.2.1657 _UART3

```
#define _UART3 _SFR(UART3_t, UART3_AddressBase)
```

UART3 struct/bit access.

Definition at line 2154 of file STM8AF_STM8S.h.

5.1.2.1658 _UART3_ADD

```
#define _UART3_ADD ((uint8_t) (0x0F << 0))
```

UART3 Address of the UART node [3:0] (in _UART3_CR4)

Definition at line 2215 of file STM8AF_STM8S.h.

5.1.2.1659 _UART3_ADD0

```
#define _UART3_ADD0 ((uint8_t) (0x01 << 0))
```

UART3 Address of the UART node [0] (in _UART3_CR4)

Definition at line 2216 of file STM8AF_STM8S.h.

5.1.2.1660 _UART3_ADD1

```
#define _UART3_ADD1 ((uint8_t) (0x01 << 1))
```

UART3 Address of the UART node [1] (in _UART3_CR4)

Definition at line 2217 of file STM8AF_STM8S.h.

5.1.2.1661 _UART3_ADD2

```
#define _UART3_ADD2 ((uint8_t) (0x01 << 2))
```

UART3 Address of the UART node [2] (in _UART3_CR4)

Definition at line 2218 of file STM8AF_STM8S.h.

5.1.2.1662 _UART3_ADD3

```
#define _UART3_ADD3 ((uint8_t) (0x01 << 3))
```

UART3 Address of the UART node [3] (in _UART3_CR4)

Definition at line 2219 of file STM8AF_STM8S.h.

5.1.2.1663 _UART3_BRR1

```
#define _UART3_BRR1 _SFR(uint8_t, UART3_AddressBase+0x02)
```

UART3 Baud rate register 1.

Definition at line 2157 of file STM8AF_STM8S.h.

5.1.2.1664 _UART3_BRR1_RESET_VALUE

```
#define _UART3_BRR1_RESET_VALUE ((uint8_t) 0x00)
```

UART3 Baud rate register 1 reset value.

Definition at line 2168 of file STM8AF_STM8S.h.

5.1.2.1665 _UART3_BRR2

```
#define _UART3_BRR2 _SFR(uint8_t, UART3_AddressBase+0x03)
```

UART3 Baud rate register 2.

Definition at line 2158 of file STM8AF_STM8S.h.

5.1.2.1666 _UART3_BRR2_RESET_VALUE

```
#define _UART3_BRR2_RESET_VALUE ((uint8_t) 0x00)
```

UART3 Baud rate register 2 reset value.

Definition at line 2169 of file STM8AF_STM8S.h.

5.1.2.1667 _UART3_CR1

```
#define _UART3_CR1 _SFR(uint8_t, UART3_AddressBase+0x04)
```

UART3 Control register 1.

Definition at line 2159 of file STM8AF_STM8S.h.

5.1.2.1668 _UART3_CR1_RESET_VALUE

```
#define _UART3_CR1_RESET_VALUE ((uint8_t) 0x00)
```

UART3 Control register 1 reset value.

Definition at line 2170 of file STM8AF_STM8S.h.

5.1.2.1669 _UART3_CR2

```
#define _UART3_CR2 _SFR(uint8_t, UART3_AddressBase+0x05)
```

UART3 Control register 2.

Definition at line 2160 of file STM8AF_STM8S.h.

5.1.2.1670 _UART3_CR2_RESET_VALUE

```
#define _UART3_CR2_RESET_VALUE ((uint8_t) 0x00)
```

UART3 Control register 2 reset value.

Definition at line 2171 of file STM8AF_STM8S.h.

5.1.2.1671 _UART3_CR3

```
#define _UART3_CR3 _SFR(uint8_t, UART3_AddressBase+0x06)
```

UART3 Control register 3.

Definition at line 2161 of file STM8AF_STM8S.h.

5.1.2.1672 _UART3_CR3_RESET_VALUE

```
#define _UART3_CR3_RESET_VALUE ((uint8_t) 0x00)
```

UART3 Control register 3 reset value.

Definition at line 2172 of file STM8AF_STM8S.h.

5.1.2.1673 _UART3_CR4

```
#define _UART3_CR4 _SFR(uint8_t, UART3_AddressBase+0x07)
```

UART3 Control register 4.

Definition at line 2162 of file STM8AF_STM8S.h.

5.1.2.1674 _UART3_CR4_RESET_VALUE

```
#define _UART3_CR4_RESET_VALUE ((uint8_t) 0x00)
```

UART3 Control register 4 reset value.

Definition at line 2173 of file STM8AF_STM8S.h.

5.1.2.1675 _UART3_CR6

```
#define _UART3_CR6 _SFR(uint8_t, UART3_AddressBase+0x09)
```

UART3 Control register 6.

Definition at line 2164 of file STM8AF_STM8S.h.

5.1.2.1676 _UART3_CR6_RESET_VALUE

```
#define _UART3_CR6_RESET_VALUE ((uint8_t) 0x00)
```

UART3 Control register 6 reset value.

Definition at line 2174 of file STM8AF_STM8S.h.

5.1.2.1677 _UART3_DR

```
#define _UART3_DR _SFR(uint8_t, UART3_AddressBase+0x01)
```

UART3 data register.

Definition at line 2156 of file STM8AF_STM8S.h.

5.1.2.1678 _UART3_FE

```
#define _UART3_FE ((uint8_t) (0x01 << 1))
```

UART3 Framing error [0] (in _UART3_SR)

Definition at line 2178 of file STM8AF_STM8S.h.

5.1.2.1679 _UART3_IDLE

```
#define _UART3_IDLE ((uint8_t) (0x01 << 4))
```

UART3 IDLE line detected [0] (in _UART3_SR)

Definition at line 2181 of file STM8AF_STM8S.h.

5.1.2.1680 _UART3_ILIEN

```
#define _UART3_ILIEN ((uint8_t) (0x01 << 4))
```

UART3 IDLE Line interrupt enable [0] (in _UART3_CR2)

Definition at line 2201 of file STM8AF_STM8S.h.

5.1.2.1681 _UART3_LASE

```
#define _UART3_LASE ((uint8_t) (0x01 << 4))
```

UART3 LIN automatic resynchronisation enable [0] (in _UART3_CR6)

Definition at line 2230 of file STM8AF_STM8S.h.

5.1.2.1682 _UART3_LBDF

```
#define _UART3_LBDF ((uint8_t) (0x01 << 4))
```

UART3 LIN Break Detection Flag [0] (in _UART3_CR4)

Definition at line 2220 of file STM8AF_STM8S.h.

5.1.2.1683 _UART3_LBDIEN

```
#define _UART3_LBDIEN ((uint8_t) (0x01 << 6))
```

UART3 LIN Break Detection Interrupt Enable [0] (in _UART3_CR4)

Definition at line 2222 of file STM8AF_STM8S.h.

5.1.2.1684 _UART3_LBDL

```
#define _UART3_LBDL ((uint8_t) (0x01 << 5))
```

UART3 LIN Break Detection Length [0] (in _UART3_CR4)

Definition at line 2221 of file STM8AF_STM8S.h.

5.1.2.1685 _UART3_LDUM

```
#define _UART3_LDUM ((uint8_t) (0x01 << 7))
```

UART3 LIN Divider Update Method [0] (in _UART3_CR6)

Definition at line 2233 of file STM8AF_STM8S.h.

5.1.2.1686 _UART3_LHDF

```
#define _UART3_LHDF ((uint8_t) (0x01 << 1))
```

UART3 LIN Header Detection Flag [0] (in _UART3_CR6)

Definition at line 2227 of file STM8AF_STM8S.h.

5.1.2.1687 _UART3_LHDIEN

```
#define _UART3_LHDIEN ((uint8_t) (0x01 << 2))
```

UART3 LIN Header Detection Interrupt Enable [0] (in _UART3_CR6)

Definition at line 2228 of file STM8AF_STM8S.h.

5.1.2.1688 _UART3_LINEN

```
#define _UART3_LINEN ((uint8_t) (0x01 << 6))
```

UART3 LIN mode enable [0] (in _UART3_CR3)

Definition at line 2211 of file STM8AF_STM8S.h.

5.1.2.1689 _UART3_LSF

```
#define _UART3_LSF ((uint8_t) (0x01 << 0))
```

UART3 LIN Sync Field [0] (in _UART3_CR6)

Definition at line 2226 of file STM8AF_STM8S.h.

5.1.2.1690 _UART3_LSLV

```
#define _UART3_LSLV ((uint8_t) (0x01 << 5))
```

UART3 LIN Slave Enable [0] (in _UART3_CR6)

Definition at line 2231 of file STM8AF_STM8S.h.

5.1.2.1691 _UART3_M

```
#define _UART3_M ((uint8_t) (0x01 << 4))
```

UART3 word length [0] (in _UART3_CR1)

Definition at line 2191 of file STM8AF_STM8S.h.

5.1.2.1692 _UART3_NF

```
#define _UART3_NF ((uint8_t) (0x01 << 2))
```

UART3 Noise flag [0] (in _UART3_SR)

Definition at line 2179 of file STM8AF_STM8S.h.

5.1.2.1693 _UART3_OR_LHE

```
#define _UART3_OR_LHE ((uint8_t) (0x01 << 3))
```

UART3 LIN Header Error (LIN slave mode) / Overrun error [0] (in _UART3_SR)

Definition at line 2180 of file STM8AF_STM8S.h.

5.1.2.1694 _UART3_PCEN

```
#define _UART3_PCEN ((uint8_t) (0x01 << 2))
```

UART3 Parity control enable [0] (in _UART3_CR1)

Definition at line 2189 of file STM8AF_STM8S.h.

5.1.2.1695 _UART3_PE

```
#define _UART3_PE ((uint8_t) (0x01 << 0))
```

UART3 Parity error [0] (in _UART3_SR)

Definition at line 2177 of file STM8AF_STM8S.h.

5.1.2.1696 _UART3_PIEN

```
#define _UART3_PIEN ((uint8_t) (0x01 << 0))
```

UART3 Parity interrupt enable [0] (in _UART3_CR1)

Definition at line 2187 of file STM8AF_STM8S.h.

5.1.2.1697 _UART3_PS

```
#define _UART3_PS ((uint8_t) (0x01 << 1))
```

UART3 Parity selection [0] (in _UART3_CR1)

Definition at line 2188 of file STM8AF_STM8S.h.

5.1.2.1698 _UART3_R8

```
#define _UART3_R8 ((uint8_t) (0x01 << 7))
```

UART3 Receive Data bit 8 (in 9-bit mode) [0] (in _UART3_CR1)

Definition at line 2194 of file STM8AF_STM8S.h.

5.1.2.1699 _UART3_REN

```
#define _UART3_REN ((uint8_t) (0x01 << 2))
```

UART3 Receiver enable [0] (in _UART3_CR2)

Definition at line 2199 of file STM8AF_STM8S.h.

5.1.2.1700 _UART3_RIEN

```
#define _UART3_RIEN ((uint8_t) (0x01 << 5))
```

UART3 Receiver interrupt enable [0] (in _UART3_CR2)

Definition at line 2202 of file STM8AF_STM8S.h.

5.1.2.1701 _UART3_RWU

```
#define _UART3_RWU ((uint8_t) (0x01 << 1))
```

UART3 Receiver wakeup [0] (in _UART3_CR2)

Definition at line 2198 of file STM8AF_STM8S.h.

5.1.2.1702 _UART3_RXNE

```
#define _UART3_RXNE ((uint8_t) (0x01 << 5))
```

UART3 Read data register not empty [0] (in _UART3_SR)

Definition at line 2182 of file STM8AF_STM8S.h.

5.1.2.1703 _UART3_SBK

```
#define _UART3_SBK ((uint8_t) (0x01 << 0))
```

UART3 Send break [0] (in _UART3_CR2)

Definition at line 2197 of file STM8AF_STM8S.h.

5.1.2.1704 _UART3_SR

```
#define _UART3_SR _SFR(uint8_t, UART3_AddressBase+0x00)
```

UART3 Status register.

Definition at line 2155 of file STM8AF_STM8S.h.

5.1.2.1705 _UART3_SR_RESET_VALUE

```
#define _UART3_SR_RESET_VALUE ((uint8_t) 0xC0)
```

UART3 Status register reset value.

Definition at line 2167 of file STM8AF_STM8S.h.

5.1.2.1706 _UART3_STOP

```
#define _UART3_STOP ((uint8_t) (0x03 << 4))
```

UART3 STOP bits [1:0] (in _UART3_CR3)

Definition at line 2208 of file STM8AF_STM8S.h.

5.1.2.1707 _UART3_STOP0

```
#define _UART3_STOP0 ((uint8_t) (0x01 << 4))
```

UART3 STOP bits [0] (in _UART3_CR3)

Definition at line 2209 of file STM8AF_STM8S.h.

5.1.2.1708 _UART3_STOP1

```
#define _UART3_STOP1 ((uint8_t) (0x01 << 5))
```

UART3 STOP bits [1] (in _UART3_CR3)

Definition at line 2210 of file STM8AF_STM8S.h.

5.1.2.1709 _UART3_T8

```
#define _UART3_T8 ((uint8_t) (0x01 << 6))
```

UART3 Transmit Data bit 8 (in 9-bit mode) [0] (in _UART3_CR1)

Definition at line 2193 of file STM8AF_STM8S.h.

5.1.2.1710 _UART3_TC

```
#define _UART3_TC ((uint8_t) (0x01 << 6))
```

UART3 Transmission complete [0] (in _UART3_SR)

Definition at line 2183 of file STM8AF_STM8S.h.

5.1.2.1711 _UART3_TCIEN

```
#define _UART3_TCIEN ((uint8_t) (0x01 << 6))
```

UART3 Transmission complete interrupt enable [0] (in _UART3_CR2)

Definition at line 2203 of file STM8AF_STM8S.h.

5.1.2.1712 _UART3_TEN

```
#define _UART3_TEN ((uint8_t) (0x01 << 3))
```

UART3 Transmitter enable [0] (in _UART3_CR2)

Definition at line 2200 of file STM8AF_STM8S.h.

5.1.2.1713 _UART3_TIEN

```
#define _UART3_TIEN ((uint8_t) (0x01 << 7))
```

UART3 Transmitter interrupt enable [0] (in _UART3_CR2)

Definition at line 2204 of file STM8AF_STM8S.h.

5.1.2.1714 _UART3_TXE

```
#define _UART3_TXE ((uint8_t) (0x01 << 7))
```

UART3 Transmit data register empty [0] (in _UART3_SR)

Definition at line 2184 of file STM8AF_STM8S.h.

5.1.2.1715 _UART3_UARTD

```
#define _UART3_UARTD ((uint8_t) (0x01 << 5))
```

UART3 Disable (for low power consumption) [0] (in _UART3_CR1)

Definition at line 2192 of file STM8AF_STM8S.h.

5.1.2.1716 _UART3_WAKE

```
#define _UART3_WAKE ((uint8_t) (0x01 << 3))
```

UART3 Wakeup method [0] (in _UART3_CR1)

Definition at line 2190 of file STM8AF_STM8S.h.

5.1.2.1717 _UART4

```
#define _UART4 _SFR(UART4_t, UART4_AddressBase)
```

UART4 struct/bit access.

Definition at line 2366 of file STM8AF_STM8S.h.

5.1.2.1718 _UART4_ADD

```
#define _UART4_ADD ((uint8_t) (0x0F << 0))
```

UART4 Address of the UART node [3:0] (in _UART4_CR4)

Definition at line 2435 of file STM8AF_STM8S.h.

5.1.2.1719 _UART4_ADD0

```
#define _UART4_ADD0 ((uint8_t) (0x01 << 0))
```

UART4 Address of the UART node [0] (in _UART4_CR4)

Definition at line 2436 of file STM8AF_STM8S.h.

5.1.2.1720 _UART4_ADD1

```
#define _UART4_ADD1 ((uint8_t) (0x01 << 1))
```

UART4 Address of the UART node [1] (in _UART4_CR4)

Definition at line 2437 of file STM8AF_STM8S.h.

5.1.2.1721 _UART4_ADD2

```
#define _UART4_ADD2 ((uint8_t) (0x01 << 2))
```

UART4 Address of the UART node [2] (in _UART4_CR4)

Definition at line 2438 of file STM8AF_STM8S.h.

5.1.2.1722 _UART4_ADD3

```
#define _UART4_ADD3 ((uint8_t) (0x01 << 3))
```

UART4 Address of the UART node [3] (in _UART4_CR4)

Definition at line 2439 of file STM8AF_STM8S.h.

5.1.2.1723 _UART4_BRR1

```
#define _UART4_BRR1 _SFR(uint8_t, UART4_AddressBase+0x02)
```

UART4 Baud rate register 1.

Definition at line 2369 of file STM8AF_STM8S.h.

5.1.2.1724 _UART4_BRR1_RESET_VALUE

```
#define _UART4_BRR1_RESET_VALUE ((uint8_t) 0x00)
```

UART4 Baud rate register 1 reset value.

Definition at line 2382 of file STM8AF_STM8S.h.

5.1.2.1725 _UART4_BRR2

```
#define _UART4_BRR2 _SFR(uint8_t, UART4_AddressBase+0x03)
```

UART4 Baud rate register 2.

Definition at line 2370 of file STM8AF_STM8S.h.

5.1.2.1726 _UART4_BRR2_RESET_VALUE

```
#define _UART4_BRR2_RESET_VALUE ((uint8_t) 0x00)
```

UART4 Baud rate register 2 reset value.

Definition at line 2383 of file STM8AF_STM8S.h.

5.1.2.1727 _UART4_CKEN

```
#define _UART4_CKEN ((uint8_t) (0x01 << 3))
```

UART4 Clock enable [0] (in _UART4_CR3)

Definition at line 2427 of file STM8AF_STM8S.h.

5.1.2.1728 _UART4_CPHA

```
#define _UART4_CPHA ((uint8_t) (0x01 << 1))
```

UART4 Clock phase [0] (in _UART4_CR3)

Definition at line 2425 of file STM8AF_STM8S.h.

5.1.2.1729 _UART4_CPOL

```
#define _UART4_CPOL ((uint8_t) (0x01 << 2))
```

UART4 Clock polarity [0] (in _UART4_CR3)

Definition at line 2426 of file STM8AF_STM8S.h.

5.1.2.1730 _UART4_CR1

```
#define _UART4_CR1 _SFR(uint8_t, UART4_AddressBase+0x04)
```

UART4 Control register 1.

Definition at line 2371 of file STM8AF_STM8S.h.

5.1.2.1731 _UART4_CR1_RESET_VALUE

```
#define _UART4_CR1_RESET_VALUE ((uint8_t) 0x00)
```

UART4 Control register 1 reset value.

Definition at line 2384 of file STM8AF_STM8S.h.

5.1.2.1732 _UART4_CR2

```
#define _UART4_CR2 _SFR(uint8_t, UART4_AddressBase+0x05)
```

UART4 Control register 2.

Definition at line 2372 of file STM8AF_STM8S.h.

5.1.2.1733 _UART4_CR2_RESET_VALUE

```
#define _UART4_CR2_RESET_VALUE ((uint8_t) 0x00)
```

UART4 Control register 2 reset value.

Definition at line 2385 of file STM8AF_STM8S.h.

5.1.2.1734 _UART4_CR3

```
#define _UART4_CR3 _SFR(uint8_t, UART4_AddressBase+0x06)
```

UART4 Control register 3.

Definition at line 2373 of file STM8AF_STM8S.h.

5.1.2.1735 _UART4_CR3_RESET_VALUE

```
#define _UART4_CR3_RESET_VALUE ((uint8_t) 0x00)
```

UART4 Control register 3 reset value.

Definition at line 2386 of file STM8AF_STM8S.h.

5.1.2.1736 _UART4_CR4

```
#define _UART4_CR4 _SFR(uint8_t, UART4_AddressBase+0x07)
```

UART4 Control register 4.

Definition at line 2374 of file STM8AF_STM8S.h.

5.1.2.1737 _UART4_CR4_RESET_VALUE

```
#define _UART4_CR4_RESET_VALUE ((uint8_t) 0x00)
```

UART4 Control register 4 reset value.

Definition at line 2387 of file STM8AF_STM8S.h.

5.1.2.1738 _UART4_CR5

```
#define _UART4_CR5 _SFR(uint8_t, UART4_AddressBase+0x08)
```

UART4 Control register 5.

Definition at line 2375 of file STM8AF_STM8S.h.

5.1.2.1739 _UART4_CR5_RESET_VALUE

```
#define _UART4_CR5_RESET_VALUE ((uint8_t) 0x00)
```

UART4 Control register 5 reset value.

Definition at line 2388 of file STM8AF_STM8S.h.

5.1.2.1740 _UART4_CR6

```
#define _UART4_CR6 _SFR(uint8_t, UART4_AddressBase+0x09)
```

UART4 Control register 6.

Definition at line 2376 of file STM8AF_STM8S.h.

5.1.2.1741 _UART4_CR6_RESET_VALUE

```
#define _UART4_CR6_RESET_VALUE ((uint8_t) 0x00)
```

UART4 Control register 6 reset value.

Definition at line 2389 of file STM8AF_STM8S.h.

5.1.2.1742 _UART4_DR

```
#define _UART4_DR _SFR(uint8_t, UART4_AddressBase+0x01)
```

UART4 data register.

Definition at line 2368 of file STM8AF_STM8S.h.

5.1.2.1743 _UART4_FE

```
#define _UART4_FE ((uint8_t) (0x01 << 1))
```

UART4 Framing error [0] (in _UART4_SR)

Definition at line 2395 of file STM8AF_STM8S.h.

5.1.2.1744 _UART4_GTR

```
#define _UART4_GTR _SFR(uint8_t, UART4_AddressBase+0x0A)
```

UART4 guard time register.

Definition at line 2377 of file STM8AF_STM8S.h.

5.1.2.1745 _UART4_GTR_RESET_VALUE

```
#define _UART4_GTR_RESET_VALUE ((uint8_t) 0x00)
```

UART4 guard time register reset value.

Definition at line 2390 of file STM8AF_STM8S.h.

5.1.2.1746 _UART4_HDSEL

```
#define _UART4_HDSEL ((uint8_t) (0x01 << 3))
```

UART4 Half-Duplex Selection [0] (in _UART4_CR5)

Definition at line 2449 of file STM8AF_STM8S.h.

5.1.2.1747 _UART4_IDLE

```
#define _UART4_IDLE ((uint8_t) (0x01 << 4))
```

UART4 IDLE line detected [0] (in _UART4_SR)

Definition at line 2398 of file STM8AF_STM8S.h.

5.1.2.1748 _UART4_ILIEN

```
#define _UART4_ILIEN ((uint8_t) (0x01 << 4))
```

UART4 IDLE Line interrupt enable [0] (in _UART4_CR2)

Definition at line 2418 of file STM8AF_STM8S.h.

5.1.2.1749 _UART4_IREN

```
#define _UART4_IREN ((uint8_t) (0x01 << 1))
```

UART4 IrDA mode Enable [0] (in _UART4_CR5)

Definition at line 2447 of file STM8AF_STM8S.h.

5.1.2.1750 _UART4_IRLP

```
#define _UART4_IRLP ((uint8_t) (0x01 << 2))
```

UART4 IrDA Low Power [0] (in _UART4_CR5)

Definition at line 2448 of file STM8AF_STM8S.h.

5.1.2.1751 _UART4_LASE

```
#define _UART4_LASE ((uint8_t) (0x01 << 4))
```

UART4 LIN automatic resynchronisation enable [0] (in _UART4_CR6)

Definition at line 2459 of file STM8AF_STM8S.h.

5.1.2.1752 _UART4_LBCL

```
#define _UART4_LBCL ((uint8_t) (0x01 << 0))
```

UART4 Last bit clock pulse [0] (in _UART4_CR3)

Definition at line 2424 of file STM8AF_STM8S.h.

5.1.2.1753 _UART4_LBDF

```
#define _UART4_LBDF ((uint8_t) (0x01 << 4))
```

UART4 LIN Break Detection Flag [0] (in _UART4_CR4)

Definition at line 2440 of file STM8AF_STM8S.h.

5.1.2.1754 _UART4_LBDIEN

```
#define _UART4_LBDIEN ((uint8_t) (0x01 << 6))
```

UART4 LIN Break Detection Interrupt Enable [0] (in _UART4_CR4)

Definition at line 2442 of file STM8AF_STM8S.h.

5.1.2.1755 _UART4_LBDL

```
#define _UART4_LBDL ((uint8_t) (0x01 << 5))
```

UART4 LIN Break Detection Length [0] (in _UART4_CR4)

Definition at line 2441 of file STM8AF_STM8S.h.

5.1.2.1756 _UART4_LDUM

```
#define _UART4_LDUM ((uint8_t) (0x01 << 7))
```

UART4 LIN Divider Update Method [0] (in _UART4_CR6)

Definition at line 2462 of file STM8AF_STM8S.h.

5.1.2.1757 _UART4_LHDF

```
#define _UART4_LHDF ((uint8_t) (0x01 << 1))
```

UART4 LIN Header Detection Flag [0] (in _UART4_CR6)

Definition at line 2456 of file STM8AF_STM8S.h.

5.1.2.1758 _UART4_LHDIEN

```
#define _UART4_LHDIEN ((uint8_t) (0x01 << 2))
```

UART4 LIN Header Detection Interrupt Enable [0] (in _UART4_CR6)

Definition at line 2457 of file STM8AF_STM8S.h.

5.1.2.1759 _UART4_LINEN

```
#define _UART4_LINEN ((uint8_t) (0x01 << 6))
```

UART4 LIN mode enable [0] (in _UART4_CR3)

Definition at line 2431 of file STM8AF_STM8S.h.

5.1.2.1760 _UART4_LSF

```
#define _UART4_LSF ((uint8_t) (0x01 << 0))
```

UART4 LIN Sync Field [0] (in _UART4_CR6)

Definition at line 2455 of file STM8AF_STM8S.h.

5.1.2.1761 _UART4_LSLV

```
#define _UART4_LSLV ((uint8_t) (0x01 << 5))
```

UART4 LIN Slave Enable [0] (in _UART4_CR6)

Definition at line 2460 of file STM8AF_STM8S.h.

5.1.2.1762 _UART4_M

```
#define _UART4_M ((uint8_t) (0x01 << 4))
```

UART4 word length [0] (in _UART4_CR1)

Definition at line 2408 of file STM8AF_STM8S.h.

5.1.2.1763 _UART4_NACK

```
#define _UART4_NACK ((uint8_t) (0x01 << 4))
```

UART4 Smartcard NACK enable [0] (in _UART4_CR5)

Definition at line 2450 of file STM8AF_STM8S.h.

5.1.2.1764 _UART4_NF

```
#define _UART4_NF ((uint8_t) (0x01 << 2))
```

UART4 Noise flag [0] (in _UART4_SR)

Definition at line 2396 of file STM8AF_STM8S.h.

5.1.2.1765 _UART4_OR_LHE

```
#define _UART4_OR_LHE ((uint8_t) (0x01 << 3))
```

UART4 LIN Header Error (LIN slave mode) / Overrun error [0] (in _UART4_SR)

Definition at line 2397 of file STM8AF_STM8S.h.

5.1.2.1766 _UART4_PCEN

```
#define _UART4_PCEN ((uint8_t) (0x01 << 2))
```

UART4 Parity control enable [0] (in _UART4_CR1)

Definition at line 2406 of file STM8AF_STM8S.h.

5.1.2.1767 _UART4_PE

```
#define _UART4_PE ((uint8_t) (0x01 << 0))
```

UART4 Parity error [0] (in _UART4_SR)

Definition at line 2394 of file STM8AF_STM8S.h.

5.1.2.1768 _UART4_PIEN

```
#define _UART4_PIEN ((uint8_t) (0x01 << 0))
```

UART4 Parity interrupt enable [0] (in _UART4_CR1)

Definition at line 2404 of file STM8AF_STM8S.h.

5.1.2.1769 _UART4_PS

```
#define _UART4_PS ((uint8_t) (0x01 << 1))
```

UART4 Parity selection [0] (in _UART4_CR1)

Definition at line 2405 of file STM8AF_STM8S.h.

5.1.2.1770 _UART4_PSCR

```
#define _UART4_PSCR _SFR(uint8_t, UART4_AddressBase+0x0B)
```

UART4 prescaler register.

Definition at line 2378 of file STM8AF_STM8S.h.

5.1.2.1771 _UART4_PSCR_RESET_VALUE

```
#define _UART4_PSCR_RESET_VALUE ((uint8_t) 0x00)
```

UART4 prescaler register reset value.

Definition at line 2391 of file STM8AF_STM8S.h.

5.1.2.1772 _UART4_R8

```
#define _UART4_R8 ((uint8_t) (0x01 << 7))
```

UART4 Receive Data bit 8 (in 9-bit mode) [0] (in _UART4_CR1)

Definition at line 2411 of file STM8AF_STM8S.h.

5.1.2.1773 _UART4_REN

```
#define _UART4_REN ((uint8_t) (0x01 << 2))
```

UART4 Receiver enable [0] (in _UART4_CR2)

Definition at line 2416 of file STM8AF_STM8S.h.

5.1.2.1774 _UART4_RIEN

```
#define _UART4_RIEN ((uint8_t) (0x01 << 5))
```

UART4 Receiver interrupt enable [0] (in _UART4_CR2)

Definition at line 2419 of file STM8AF_STM8S.h.

5.1.2.1775 _UART4_RWU

```
#define _UART4_RWU ((uint8_t) (0x01 << 1))
```

UART4 Receiver wakeup [0] (in _UART4_CR2)

Definition at line 2415 of file STM8AF_STM8S.h.

5.1.2.1776 _UART4_RXNE

```
#define _UART4_RXNE ((uint8_t) (0x01 << 5))
```

UART4 Read data register not empty [0] (in _UART4_SR)

Definition at line 2399 of file STM8AF_STM8S.h.

5.1.2.1777 _UART4_SBK

```
#define _UART4_SBK ((uint8_t) (0x01 << 0))
```

UART4 Send break [0] (in _UART4_CR2)

Definition at line 2414 of file STM8AF_STM8S.h.

5.1.2.1778 _UART4_SCEN

```
#define _UART4_SCEN ((uint8_t) (0x01 << 5))
```

UART4 Smartcard mode enable [0] (in _UART4_CR5)

Definition at line 2451 of file STM8AF_STM8S.h.

5.1.2.1779 _UART4_SR

```
#define _UART4_SR _SFR(uint8_t, UART4_AddressBase+0x00)
```

UART4 Status register.

Definition at line 2367 of file STM8AF_STM8S.h.

5.1.2.1780 _UART4_SR_RESET_VALUE

```
#define _UART4_SR_RESET_VALUE ((uint8_t) 0xC0)
```

UART4 Status register reset value.

Definition at line 2381 of file STM8AF_STM8S.h.

5.1.2.1781 _UART4_STOP

```
#define _UART4_STOP ((uint8_t) (0x03 << 4))
```

UART4 STOP bits [1:0] (in _UART4_CR3)

Definition at line 2428 of file STM8AF_STM8S.h.

5.1.2.1782 _UART4_STOP0

```
#define _UART4_STOP0 ((uint8_t) (0x01 << 4))
```

UART4 STOP bits [0] (in _UART4_CR3)

Definition at line 2429 of file STM8AF_STM8S.h.

5.1.2.1783 _UART4_STOP1

```
#define _UART4_STOP1 ((uint8_t) (0x01 << 5))
```

UART4 STOP bits [1] (in _UART4_CR3)

Definition at line 2430 of file STM8AF_STM8S.h.

5.1.2.1784 _UART4_T8

```
#define _UART4_T8 ((uint8_t) (0x01 << 6))
```

UART4 Transmit Data bit 8 (in 9-bit mode) [0] (in _UART4_CR1)

Definition at line 2410 of file STM8AF_STM8S.h.

5.1.2.1785 _UART4_TC

```
#define _UART4_TC ((uint8_t) (0x01 << 6))
```

UART4 Transmission complete [0] (in _UART4_SR)

Definition at line 2400 of file STM8AF_STM8S.h.

5.1.2.1786 _UART4_TCIEN

```
#define _UART4_TCIEN ((uint8_t) (0x01 << 6))
```

UART4 Transmission complete interrupt enable [0] (in _UART4_CR2)

Definition at line 2420 of file STM8AF_STM8S.h.

5.1.2.1787 _UART4_TEN

```
#define _UART4_TEN ((uint8_t) (0x01 << 3))
```

UART4 Transmitter enable [0] (in _UART4_CR2)

Definition at line 2417 of file STM8AF_STM8S.h.

5.1.2.1788 _UART4_TIEN

```
#define _UART4_TIEN ((uint8_t) (0x01 << 7))
```

UART4 Transmitter interrupt enable [0] (in _UART4_CR2)

Definition at line 2421 of file STM8AF_STM8S.h.

5.1.2.1789 _UART4_TXE

```
#define _UART4_TXE ((uint8_t) (0x01 << 7))
```

UART4 Transmit data register empty [0] (in _UART4_SR)

Definition at line 2401 of file STM8AF_STM8S.h.

5.1.2.1790 _UART4_UARTD

```
#define _UART4_UARTD ((uint8_t) (0x01 << 5))
```

UART4 Disable (for low power consumption) [0] (in _UART4_CR1)

Definition at line 2409 of file STM8AF_STM8S.h.

5.1.2.1791 _UART4_WAKE

```
#define _UART4_WAKE ((uint8_t) (0x01 << 3))
```

UART4 Wakeup method [0] (in _UART4_CR1)

Definition at line 2407 of file STM8AF_STM8S.h.

5.1.2.1792 _WWDG

```
#define _WWDG _SFR(WWDG_t, WWDG_AddressBase)
```

Window Watchdog struct/bit access.

Definition at line 1006 of file STM8AF_STM8S.h.

5.1.2.1793 _WWDG_CR

```
#define _WWDG_CR _SFR(uint8_t, WWDG_AddressBase+0x00)
```

Window Watchdog Control register (WWDG_CR)

Definition at line 1007 of file STM8AF_STM8S.h.

5.1.2.1794 _WWDG_CR_RESET_VALUE

```
#define _WWDG_CR_RESET_VALUE ((uint8_t) 0x7F)
```

Window Watchdog Control register reset value.

Definition at line 1011 of file STM8AF_STM8S.h.

5.1.2.1795 _WWDG_T

```
#define _WWDG_T ((uint8_t) (0x7F << 0))
```

Window Watchdog 7-bit counter [6:0] (in _WWDG_CR)

Definition at line 1015 of file STM8AF_STM8S.h.

5.1.2.1796 _WWDG_T0

```
#define _WWDG_T0 ((uint8_t) (0x01 << 0))
```

Window Watchdog 7-bit counter [0] (in _WWDG_CR)

Definition at line 1016 of file STM8AF_STM8S.h.

5.1.2.1797 _WWDG_T1

```
#define _WWDG_T1 ((uint8_t) (0x01 << 1))
```

Window Watchdog 7-bit counter [1] (in _WWDG_CR)

Definition at line 1017 of file STM8AF_STM8S.h.

5.1.2.1798 _WWDG_T2

```
#define _WWDG_T2 ((uint8_t) (0x01 << 2))
```

Window Watchdog 7-bit counter [2] (in _WWDG_CR)

Definition at line 1018 of file STM8AF_STM8S.h.

5.1.2.1799 _WWDG_T3

```
#define _WWDG_T3 ((uint8_t) (0x01 << 3))
```

Window Watchdog 7-bit counter [3] (in _WWDG_CR)

Definition at line 1019 of file STM8AF_STM8S.h.

5.1.2.1800 _WWDG_T4

```
#define _WWDG_T4 ((uint8_t) (0x01 << 4))
```

Window Watchdog 7-bit counter [4] (in _WWDG_CR)

Definition at line 1020 of file STM8AF_STM8S.h.

5.1.2.1801 _WWDG_T5

```
#define _WWDG_T5 ((uint8_t) (0x01 << 5))
```

Window Watchdog 7-bit counter [5] (in _WWDG_CR)

Definition at line 1021 of file STM8AF_STM8S.h.

5.1.2.1802 _WWDG_T6

```
#define _WWDG_T6 ((uint8_t) (0x01 << 6))
```

Window Watchdog 7-bit counter [6] (in _WWDG_CR)

Definition at line 1022 of file STM8AF_STM8S.h.

5.1.2.1803 _WWDG_W

```
#define _WWDG_W ((uint8_t) (0x7F << 0))
```

Window Watchdog 7-bit window value [6:0] (in _WWDG_WR)

Definition at line 1026 of file STM8AF_STM8S.h.

5.1.2.1804 _WWDG_W0

```
#define _WWDG_W0 ((uint8_t) (0x01 << 0))
```

Window Watchdog 7-bit window value [0] (in _WWDG_WR)

Definition at line 1027 of file STM8AF_STM8S.h.

5.1.2.1805 _WWDG_W1

```
#define _WWDG_W1 ((uint8_t) (0x01 << 1))
```

Window Watchdog 7-bit window value [1] (in _WWDG_WR)

Definition at line 1028 of file STM8AF_STM8S.h.

5.1.2.1806 _WWDG_W2

```
#define _WWDG_W2 ((uint8_t) (0x01 << 2))
```

Window Watchdog 7-bit window value [2] (in _WWDG_WR)

Definition at line 1029 of file STM8AF_STM8S.h.

5.1.2.1807 _WWDG_W3

```
#define _WWDG_W3 ((uint8_t) (0x01 << 3))
```

Window Watchdog 7-bit window value [3] (in _WWDG_WR)

Definition at line 1030 of file STM8AF_STM8S.h.

5.1.2.1808 _WWDG_W4

```
#define _WWDG_W4 ((uint8_t) (0x01 << 4))
```

Window Watchdog 7-bit window value [4] (in _WWDG_WR)

Definition at line 1031 of file STM8AF_STM8S.h.

5.1.2.1809 _WWDG_W5

```
#define _WWDG_W5 ((uint8_t) (0x01 << 5))
```

Window Watchdog 7-bit window value [5] (in _WWDG_WR)

Definition at line 1032 of file STM8AF_STM8S.h.

5.1.2.1810 _WWDG_W6

```
#define _WWDG_W6 ((uint8_t) (0x01 << 6))
```

Window Watchdog 7-bit window value [6] (in _WWDG_WR)

Definition at line 1033 of file STM8AF_STM8S.h.

5.1.2.1811 _WWDG_WDGA

```
#define _WWDG_WDGA ((uint8_t) (0x01 << 7))
```

Window Watchdog activation bit (n/a if WWDG enabled by option byte) [0] (in _WWDG_CR)

Definition at line 1023 of file STM8AF_STM8S.h.

5.1.2.1812 _WWDG_WR

```
#define _WWDG_WR _SFR(uint8_t, WWDG_AddressBase+0x01)
```

Window Watchdog Window register (WWDG_WR)

Definition at line 1008 of file STM8AF_STM8S.h.

5.1.2.1813 _WWDG_WR_RESET_VALUE

```
#define _WWDG_WR_RESET_VALUE ((uint8_t) 0x7F)
```

Window Watchdog Window register reset value.

Definition at line 1012 of file STM8AF_STM8S.h.

5.1.2.1814 ADC1_AddressBase [1/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8AF6213.h.

5.1.2.1815 ADC1_AddressBase [2/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8S903F3.h.

5.1.2.1816 ADC1_AddressBase [3/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8S903K3.h.

5.1.2.1817 ADC1_AddressBase [4/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8S001J3.h.

5.1.2.1818 ADC1_AddressBase [5/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8S003F3.h.

5.1.2.1819 ADC1_AddressBase [6/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8S003K3.h.

5.1.2.1820 ADC1_AddressBase [7/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8S103F2.h.

5.1.2.1821 ADC1_AddressBase [8/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8AF6213A.h.

5.1.2.1822 ADC1_AddressBase [9/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8AF6223.h.

5.1.2.1823 ADC1_AddressBase [10/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8S103F3.h.

5.1.2.1824 ADC1_AddressBase [11/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8AF6223A.h.

5.1.2.1825 ADC1_AddressBase [12/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8AF6226.h.

5.1.2.1826 ADC1_AddressBase [13/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 73 of file STM8S103K3.h.

5.1.2.1827 ADC1_AddressBase [14/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 74 of file STM8AF6366.h.

5.1.2.1828 ADC1_AddressBase [15/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8AF6246.h.

5.1.2.1829 ADC1_AddressBase [16/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8AF6248.h.

5.1.2.1830 ADC1_AddressBase [17/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8AF6266.h.

5.1.2.1831 ADC1_AddressBase [18/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8AF6268.h.

5.1.2.1832 ADC1_AddressBase [19/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8AF6269.h.

5.1.2.1833 ADC1_AddressBase [20/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8S005C6.h.

5.1.2.1834 ADC1_AddressBase [21/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8S005K6.h.

5.1.2.1835 ADC1_AddressBase [22/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8S105C4.h.

5.1.2.1836 ADC1_AddressBase [23/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8S105C6.h.

5.1.2.1837 ADC1_AddressBase [24/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8S105K4.h.

5.1.2.1838 ADC1_AddressBase [25/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8S105K6.h.

5.1.2.1839 ADC1_AddressBase [26/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8S105S4.h.

5.1.2.1840 ADC1_AddressBase [27/27]

```
#define ADC1_AddressBase 0x53E0
```

Definition at line 75 of file STM8S105S6.h.

5.1.2.1841 ADC2_AddressBase [1/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF5269.h.

5.1.2.1842 ADC2_AddressBase [2/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S208S8.h.

5.1.2.1843 ADC2_AddressBase [3/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF52A8.h.

5.1.2.1844 ADC2_AddressBase [4/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S208SB.h.

5.1.2.1845 ADC2_AddressBase [5/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF5288.h.

5.1.2.1846 ADC2_AddressBase [6/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S208R8.h.

5.1.2.1847 ADC2_AddressBase [7/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF52A9.h.

5.1.2.1848 ADC2_AddressBase [8/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF5289.h.

5.1.2.1849 ADC2_AddressBase [9/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S208CB.h.

5.1.2.1850 ADC2_AddressBase [10/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF6286.h.

5.1.2.1851 ADC2_AddressBase [11/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF52AA.h.

5.1.2.1852 ADC2_AddressBase [12/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF6288.h.

5.1.2.1853 ADC2_AddressBase [13/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S007C8.h.

5.1.2.1854 ADC2_AddressBase [14/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF6289.h.

5.1.2.1855 ADC2_AddressBase [15/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF528A.h.

5.1.2.1856 ADC2_AddressBase [16/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF628A.h.

5.1.2.1857 ADC2_AddressBase [17/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF62A6.h.

5.1.2.1858 ADC2_AddressBase [18/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF5268.h.

5.1.2.1859 ADC2_AddressBase [19/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207C6.h.

5.1.2.1860 ADC2_AddressBase [20/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207C8.h.

5.1.2.1861 ADC2_AddressBase [21/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207CB.h.

5.1.2.1862 ADC2_AddressBase [22/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF5286.h.

5.1.2.1863 ADC2_AddressBase [23/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF62A8.h.

5.1.2.1864 ADC2_AddressBase [24/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207K6.h.

5.1.2.1865 ADC2_AddressBase [25/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207K8.h.

5.1.2.1866 ADC2_AddressBase [26/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207M8.h.

5.1.2.1867 ADC2_AddressBase [27/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207MB.h.

5.1.2.1868 ADC2_AddressBase [28/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF62A9.h.

5.1.2.1869 ADC2_AddressBase [29/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF52A6.h.

5.1.2.1870 ADC2_AddressBase [30/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207R6.h.

5.1.2.1871 ADC2_AddressBase [31/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207R8.h.

5.1.2.1872 ADC2_AddressBase [32/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207RB.h.

5.1.2.1873 ADC2_AddressBase [33/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF62AA.h.

5.1.2.1874 ADC2_AddressBase [34/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207S6.h.

5.1.2.1875 ADC2_AddressBase [35/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207S8.h.

5.1.2.1876 ADC2_AddressBase [36/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S207SB.h.

5.1.2.1877 ADC2_AddressBase [37/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S208C6.h.

5.1.2.1878 ADC2_AddressBase [38/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S208C8.h.

5.1.2.1879 ADC2_AddressBase [39/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S208MB.h.

5.1.2.1880 ADC2_AddressBase [40/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8AF6388.h.

5.1.2.1881 ADC2_AddressBase [41/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S208R6.h.

5.1.2.1882 ADC2_AddressBase [42/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S208RB.h.

5.1.2.1883 ADC2_AddressBase [43/43]

```
#define ADC2_AddressBase 0x5400
```

Definition at line 78 of file STM8S208S6.h.

5.1.2.1884 AWU_AddressBase [1/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8S103K3.h.

5.1.2.1885 AWU_AddressBase [2/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8S903F3.h.

5.1.2.1886 AWU_AddressBase [3/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8S903K3.h.

5.1.2.1887 AWU_AddressBase [4/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8S001J3.h.

5.1.2.1888 AWU_AddressBase [5/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8S003F3.h.

5.1.2.1889 AWU_AddressBase [6/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8S003K3.h.

5.1.2.1890 AWU_AddressBase [7/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8AF6213.h.

5.1.2.1891 AWU_AddressBase [8/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8S103F2.h.

5.1.2.1892 AWU_AddressBase [9/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8S103F3.h.

5.1.2.1893 AWU_AddressBase [10/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8AF6213A.h.

5.1.2.1894 AWU_AddressBase [11/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8AF6223.h.

5.1.2.1895 AWU_AddressBase [12/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8AF6223A.h.

5.1.2.1896 AWU_AddressBase [13/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8AF6366.h.

5.1.2.1897 AWU_AddressBase [14/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 65 of file STM8AF6226.h.

5.1.2.1898 AWU_AddressBase [15/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8AF6246.h.

5.1.2.1899 AWU_AddressBase [16/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8S005K6.h.

5.1.2.1900 AWU_AddressBase [17/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8AF6248.h.

5.1.2.1901 AWU_AddressBase [18/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8AF6266.h.

5.1.2.1902 AWU_AddressBase [19/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8AF6268.h.

5.1.2.1903 AWU_AddressBase [20/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8AF6269.h.

5.1.2.1904 AWU_AddressBase [21/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8S005C6.h.

5.1.2.1905 AWU_AddressBase [22/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8S105C6.h.

5.1.2.1906 AWU_AddressBase [23/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8S105K4.h.

5.1.2.1907 AWU_AddressBase [24/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8S105K6.h.

5.1.2.1908 AWU_AddressBase [25/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8S105S4.h.

5.1.2.1909 AWU_AddressBase [26/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8S105S6.h.

5.1.2.1910 AWU_AddressBase [27/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 66 of file STM8S105C4.h.

5.1.2.1911 AWU_AddressBase [28/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S208S8.h.

5.1.2.1912 AWU_AddressBase [29/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S208SB.h.

5.1.2.1913 AWU_AddressBase [30/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF52A8.h.

5.1.2.1914 AWU_AddressBase [31/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207R8.h.

5.1.2.1915 AWU_AddressBase [32/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF5288.h.

5.1.2.1916 AWU_AddressBase [33/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF5268.h.

5.1.2.1917 AWU_AddressBase [34/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207S6.h.

5.1.2.1918 AWU_AddressBase [35/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF52A9.h.

5.1.2.1919 AWU_AddressBase [36/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207CB.h.

5.1.2.1920 AWU_AddressBase [37/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF5289.h.

5.1.2.1921 AWU_AddressBase [38/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S208S6.h.

5.1.2.1922 AWU_AddressBase [39/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S208R6.h.

5.1.2.1923 AWU_AddressBase [40/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF52AA.h.

5.1.2.1924 AWU_AddressBase [41/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF6286.h.

5.1.2.1925 AWU_AddressBase [42/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207MB.h.

5.1.2.1926 AWU_AddressBase [43/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF5286.h.

5.1.2.1927 AWU_AddressBase [44/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207K8.h.

5.1.2.1928 AWU_AddressBase [45/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF6288.h.

5.1.2.1929 AWU_AddressBase [46/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF6289.h.

5.1.2.1930 AWU_AddressBase [47/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S007C8.h.

5.1.2.1931 AWU_AddressBase [48/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF528A.h.

5.1.2.1932 AWU_AddressBase [49/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF628A.h.

5.1.2.1933 AWU_AddressBase [50/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF5269.h.

5.1.2.1934 AWU_AddressBase [51/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF62A6.h.

5.1.2.1935 AWU_AddressBase [52/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207C8.h.

5.1.2.1936 AWU_AddressBase [53/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207K6.h.

5.1.2.1937 AWU_AddressBase [54/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF62A8.h.

5.1.2.1938 AWU_AddressBase [55/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207M8.h.

5.1.2.1939 AWU_AddressBase [56/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207R6.h.

5.1.2.1940 AWU_AddressBase [57/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF62A9.h.

5.1.2.1941 AWU_AddressBase [58/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207RB.h.

5.1.2.1942 AWU_AddressBase [59/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF62AA.h.

5.1.2.1943 AWU_AddressBase [60/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207S8.h.

5.1.2.1944 AWU_AddressBase [61/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207SB.h.

5.1.2.1945 AWU_AddressBase [62/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S208C6.h.

5.1.2.1946 AWU_AddressBase [63/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S208C8.h.

5.1.2.1947 AWU_AddressBase [64/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S208CB.h.

5.1.2.1948 AWU_AddressBase [65/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S207C6.h.

5.1.2.1949 AWU_AddressBase [66/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S208MB.h.

5.1.2.1950 AWU_AddressBase [67/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF6388.h.

5.1.2.1951 AWU_AddressBase [68/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S208R8.h.

5.1.2.1952 AWU_AddressBase [69/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8AF52A6.h.

5.1.2.1953 AWU_AddressBase [70/70]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 68 of file STM8S208RB.h.

5.1.2.1954 BEEP_AddressBase [1/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8S903F3.h.

5.1.2.1955 BEEP_AddressBase [2/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8S903K3.h.

5.1.2.1956 BEEP_AddressBase [3/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8AF6223A.h.

5.1.2.1957 BEEP_AddressBase [4/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8S001J3.h.

5.1.2.1958 BEEP_AddressBase [5/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8S003F3.h.

5.1.2.1959 BEEP_AddressBase [6/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8S003K3.h.

5.1.2.1960 BEEP_AddressBase [7/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8AF6213.h.

5.1.2.1961 BEEP_AddressBase [8/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8S103F2.h.

5.1.2.1962 BEEP_AddressBase [9/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8S103F3.h.

5.1.2.1963 BEEP_AddressBase [10/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8S103K3.h.

5.1.2.1964 BEEP_AddressBase [11/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8AF6213A.h.

5.1.2.1965 BEEP_AddressBase [12/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8AF6223.h.

5.1.2.1966 BEEP_AddressBase [13/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8AF6366.h.

5.1.2.1967 BEEP_AddressBase [14/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 66 of file STM8AF6226.h.

5.1.2.1968 BEEP_AddressBase [15/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8AF6246.h.

5.1.2.1969 BEEP_AddressBase [16/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8AF6248.h.

5.1.2.1970 BEEP_AddressBase [17/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8AF6266.h.

5.1.2.1971 BEEP_AddressBase [18/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8AF6268.h.

5.1.2.1972 BEEP_AddressBase [19/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8AF6269.h.

5.1.2.1973 BEEP_AddressBase [20/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8S005C6.h.

5.1.2.1974 BEEP_AddressBase [21/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8S005K6.h.

5.1.2.1975 BEEP_AddressBase [22/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8S105C4.h.

5.1.2.1976 BEEP_AddressBase [23/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8S105C6.h.

5.1.2.1977 BEEP_AddressBase [24/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8S105K4.h.

5.1.2.1978 BEEP_AddressBase [25/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8S105K6.h.

5.1.2.1979 BEEP_AddressBase [26/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8S105S4.h.

5.1.2.1980 BEEP_AddressBase [27/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 67 of file STM8S105S6.h.

5.1.2.1981 BEEP_AddressBase [28/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S208S8.h.

5.1.2.1982 BEEP_AddressBase [29/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF5288.h.

5.1.2.1983 BEEP_AddressBase [30/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF5268.h.

5.1.2.1984 BEEP_AddressBase [31/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF52A8.h.

5.1.2.1985 BEEP_AddressBase [32/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S208SB.h.

5.1.2.1986 BEEP_AddressBase [33/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF52A9.h.

5.1.2.1987 BEEP_AddressBase [34/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S208CB.h.

5.1.2.1988 BEEP_AddressBase [35/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S208R8.h.

5.1.2.1989 BEEP_AddressBase [36/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF5289.h.

5.1.2.1990 BEEP_AddressBase [37/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S208S6.h.

5.1.2.1991 BEEP_AddressBase [38/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S208MB.h.

5.1.2.1992 BEEP_AddressBase [39/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF52AA.h.

5.1.2.1993 BEEP_AddressBase [40/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF6286.h.

5.1.2.1994 BEEP_AddressBase [41/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF5286.h.

5.1.2.1995 BEEP_AddressBase [42/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF6288.h.

5.1.2.1996 BEEP_AddressBase [43/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S007C8.h.

5.1.2.1997 BEEP_AddressBase [44/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF6289.h.

5.1.2.1998 BEEP_AddressBase [45/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF628A.h.

5.1.2.1999 BEEP_AddressBase [46/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF528A.h.

5.1.2.2000 BEEP_AddressBase [47/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF62A6.h.

5.1.2.2001 BEEP_AddressBase [48/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207C6.h.

5.1.2.2002 BEEP_AddressBase [49/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207C8.h.

5.1.2.2003 BEEP_AddressBase [50/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207CB.h.

5.1.2.2004 BEEP_AddressBase [51/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207K6.h.

5.1.2.2005 BEEP_AddressBase [52/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF62A8.h.

5.1.2.2006 BEEP_AddressBase [53/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207K8.h.

5.1.2.2007 BEEP_AddressBase [54/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207M8.h.

5.1.2.2008 BEEP_AddressBase [55/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207MB.h.

5.1.2.2009 BEEP_AddressBase [56/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF62A9.h.

5.1.2.2010 BEEP_AddressBase [57/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207R6.h.

5.1.2.2011 BEEP_AddressBase [58/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207R8.h.

5.1.2.2012 BEEP_AddressBase [59/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207RB.h.

5.1.2.2013 BEEP_AddressBase [60/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207S6.h.

5.1.2.2014 BEEP_AddressBase [61/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF52A6.h.

5.1.2.2015 BEEP_AddressBase [62/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207S8.h.

5.1.2.2016 BEEP_AddressBase [63/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF5269.h.

5.1.2.2017 BEEP_AddressBase [64/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF62AA.h.

5.1.2.2018 BEEP_AddressBase [65/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S207SB.h.

5.1.2.2019 BEEP_AddressBase [66/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S208C6.h.

5.1.2.2020 BEEP_AddressBase [67/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S208C8.h.

5.1.2.2021 BEEP_AddressBase [68/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S208R6.h.

5.1.2.2022 BEEP_AddressBase [69/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8AF6388.h.

5.1.2.2023 BEEP_AddressBase [70/70]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 69 of file STM8S208RB.h.

5.1.2.2024 CAN_AddressBase [1/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8AF5269.h.

5.1.2.2025 CAN_AddressBase [2/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8AF5288.h.

5.1.2.2026 CAN_AddressBase [3/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8AF5268.h.

5.1.2.2027 CAN_AddressBase [4/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8AF52A8.h.

5.1.2.2028 CAN_AddressBase [5/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8S208SB.h.

5.1.2.2029 CAN_AddressBase [6/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8S208S8.h.

5.1.2.2030 CAN_AddressBase [7/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8S208R8.h.

5.1.2.2031 CAN_AddressBase [8/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8AF52A9.h.

5.1.2.2032 CAN_AddressBase [9/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8S208MB.h.

5.1.2.2033 CAN_AddressBase [10/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8S208C8.h.

5.1.2.2034 CAN_AddressBase [11/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8S208S6.h.

5.1.2.2035 CAN_AddressBase [12/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8S208C6.h.

5.1.2.2036 CAN_AddressBase [13/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8AF52AA.h.

5.1.2.2037 CAN_AddressBase [14/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8AF5289.h.

5.1.2.2038 CAN_AddressBase [15/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8AF5286.h.

5.1.2.2039 CAN_AddressBase [16/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8AF528A.h.

5.1.2.2040 CAN_AddressBase [17/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8S208CB.h.

5.1.2.2041 CAN_AddressBase [18/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8S208R6.h.

5.1.2.2042 CAN_AddressBase [19/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8AF52A6.h.

5.1.2.2043 CAN_AddressBase [20/20]

```
#define CAN_AddressBase 0x5420
```

Definition at line 79 of file STM8S208RB.h.

5.1.2.2044 CFG_AddressBase [1/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8S903F3.h.

5.1.2.2045 CFG_AddressBase [2/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8S903K3.h.

5.1.2.2046 CFG_AddressBase [3/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8S001J3.h.

5.1.2.2047 CFG_AddressBase [4/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8S003F3.h.

5.1.2.2048 CFG_AddressBase [5/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8S003K3.h.

5.1.2.2049 CFG_AddressBase [6/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8S103F2.h.

5.1.2.2050 CFG_AddressBase [7/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8AF6213.h.

5.1.2.2051 CFG_AddressBase [8/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8S103K3.h.

5.1.2.2052 CFG_AddressBase [9/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8AF6213A.h.

5.1.2.2053 CFG_AddressBase [10/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8AF6223.h.

5.1.2.2054 CFG_AddressBase [11/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8AF6223A.h.

5.1.2.2055 CFG_AddressBase [12/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8S103F3.h.

5.1.2.2056 CFG_AddressBase [13/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 74 of file STM8AF6226.h.

5.1.2.2057 CFG_AddressBase [14/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 75 of file STM8AF6366.h.

5.1.2.2058 CFG_AddressBase [15/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8AF6246.h.

5.1.2.2059 CFG_AddressBase [16/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8S005K6.h.

5.1.2.2060 CFG_AddressBase [17/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8S005C6.h.

5.1.2.2061 CFG_AddressBase [18/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8AF6248.h.

5.1.2.2062 CFG_AddressBase [19/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8AF6266.h.

5.1.2.2063 CFG_AddressBase [20/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8AF6268.h.

5.1.2.2064 CFG_AddressBase [21/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8AF6269.h.

5.1.2.2065 CFG_AddressBase [22/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8S105C6.h.

5.1.2.2066 CFG_AddressBase [23/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8S105K6.h.

5.1.2.2067 CFG_AddressBase [24/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8S105C4.h.

5.1.2.2068 CFG_AddressBase [25/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8S105S4.h.

5.1.2.2069 CFG_AddressBase [26/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8S105S6.h.

5.1.2.2070 CFG_AddressBase [27/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 76 of file STM8S105K4.h.

5.1.2.2071 CFG_AddressBase [28/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207S8.h.

5.1.2.2072 CFG_AddressBase [29/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8AF6286.h.

5.1.2.2073 CFG_AddressBase [30/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207K6.h.

5.1.2.2074 CFG_AddressBase [31/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207M8.h.

5.1.2.2075 CFG_AddressBase [32/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207R6.h.

5.1.2.2076 CFG_AddressBase [33/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8AF6288.h.

5.1.2.2077 CFG_AddressBase [34/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S007C8.h.

5.1.2.2078 CFG_AddressBase [35/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8AF628A.h.

5.1.2.2079 CFG_AddressBase [36/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8AF62A6.h.

5.1.2.2080 CFG_AddressBase [37/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207C6.h.

5.1.2.2081 CFG_AddressBase [38/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207C8.h.

5.1.2.2082 CFG_AddressBase [39/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207CB.h.

5.1.2.2083 CFG_AddressBase [40/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8AF62A8.h.

5.1.2.2084 CFG_AddressBase [41/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207K8.h.

5.1.2.2085 CFG_AddressBase [42/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207MB.h.

5.1.2.2086 CFG_AddressBase [43/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207R8.h.

5.1.2.2087 CFG_AddressBase [44/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8AF62A9.h.

5.1.2.2088 CFG_AddressBase [45/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207RB.h.

5.1.2.2089 CFG_AddressBase [46/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207S6.h.

5.1.2.2090 CFG_AddressBase [47/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8AF62AA.h.

5.1.2.2091 CFG_AddressBase [48/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8S207SB.h.

5.1.2.2092 CFG_AddressBase [49/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8AF6388.h.

5.1.2.2093 CFG_AddressBase [50/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 79 of file STM8AF6289.h.

5.1.2.2094 CFG_AddressBase [51/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8S208SB.h.

5.1.2.2095 CFG_AddressBase [52/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8AF5268.h.

5.1.2.2096 CFG_AddressBase [53/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8AF5269.h.

5.1.2.2097 CFG_AddressBase [54/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8AF5288.h.

5.1.2.2098 CFG_AddressBase [55/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8AF52A8.h.

5.1.2.2099 CFG_AddressBase [56/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8S208S8.h.

5.1.2.2100 CFG_AddressBase [57/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8S208R6.h.

5.1.2.2101 CFG_AddressBase [58/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8AF52A9.h.

5.1.2.2102 CFG_AddressBase [59/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8S208RB.h.

5.1.2.2103 CFG_AddressBase [60/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8AF5286.h.

5.1.2.2104 CFG_AddressBase [61/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8S208C6.h.

5.1.2.2105 CFG_AddressBase [62/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8AF5289.h.

5.1.2.2106 CFG_AddressBase [63/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8AF52AA.h.

5.1.2.2107 CFG_AddressBase [64/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8AF528A.h.

5.1.2.2108 CFG_AddressBase [65/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8S208C8.h.

5.1.2.2109 CFG_AddressBase [66/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8S208CB.h.

5.1.2.2110 CFG_AddressBase [67/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8S208MB.h.

5.1.2.2111 CFG_AddressBase [68/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8AF52A6.h.

5.1.2.2112 CFG_AddressBase [69/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8S208R8.h.

5.1.2.2113 CFG_AddressBase [70/70]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 80 of file STM8S208S6.h.

5.1.2.2114 CLK_AddressBase [1/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8S103F2.h.

5.1.2.2115 CLK_AddressBase [2/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8S903F3.h.

5.1.2.2116 CLK_AddressBase [3/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8S903K3.h.

5.1.2.2117 CLK_AddressBase [4/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8S001J3.h.

5.1.2.2118 CLK_AddressBase [5/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8S003F3.h.

5.1.2.2119 CLK_AddressBase [6/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8S003K3.h.

5.1.2.2120 CLK_AddressBase [7/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8AF6213.h.

5.1.2.2121 CLK_AddressBase [8/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8S103F3.h.

5.1.2.2122 CLK_AddressBase [9/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8S103K3.h.

5.1.2.2123 CLK_AddressBase [10/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8AF6213A.h.

5.1.2.2124 CLK_AddressBase [11/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8AF6223.h.

5.1.2.2125 CLK_AddressBase [12/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8AF6223A.h.

5.1.2.2126 CLK_AddressBase [13/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8AF6366.h.

5.1.2.2127 CLK_AddressBase [14/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 62 of file STM8AF6226.h.

5.1.2.2128 CLK_AddressBase [15/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8AF6246.h.

5.1.2.2129 CLK_AddressBase [16/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8AF6248.h.

5.1.2.2130 CLK_AddressBase [17/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8AF6266.h.

5.1.2.2131 CLK_AddressBase [18/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8AF6268.h.

5.1.2.2132 CLK_AddressBase [19/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8AF6269.h.

5.1.2.2133 CLK_AddressBase [20/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8S005C6.h.

5.1.2.2134 CLK_AddressBase [21/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8S005K6.h.

5.1.2.2135 CLK_AddressBase [22/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8S105C4.h.

5.1.2.2136 CLK_AddressBase [23/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8S105C6.h.

5.1.2.2137 CLK_AddressBase [24/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8S105S4.h.

5.1.2.2138 CLK_AddressBase [25/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8S105S6.h.

5.1.2.2139 CLK_AddressBase [26/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8S105K4.h.

5.1.2.2140 CLK_AddressBase [27/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 63 of file STM8S105K6.h.

5.1.2.2141 CLK_AddressBase [28/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S208S8.h.

5.1.2.2142 CLK_AddressBase [29/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S208SB.h.

5.1.2.2143 CLK_AddressBase [30/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF52A8.h.

5.1.2.2144 CLK_AddressBase [31/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF6289.h.

5.1.2.2145 CLK_AddressBase [32/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF52A9.h.

5.1.2.2146 CLK_AddressBase [33/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF5289.h.

5.1.2.2147 CLK_AddressBase [34/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S208MB.h.

5.1.2.2148 CLK_AddressBase [35/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF6286.h.

5.1.2.2149 CLK_AddressBase [36/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF52AA.h.

5.1.2.2150 CLK_AddressBase [37/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF6288.h.

5.1.2.2151 CLK_AddressBase [38/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF5268.h.

5.1.2.2152 CLK_AddressBase [39/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF5286.h.

5.1.2.2153 CLK_AddressBase [40/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S208RB.h.

5.1.2.2154 CLK_AddressBase [41/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF528A.h.

5.1.2.2155 CLK_AddressBase [42/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF62A6.h.

5.1.2.2156 CLK_AddressBase [43/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207C6.h.

5.1.2.2157 CLK_AddressBase [44/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207C8.h.

5.1.2.2158 CLK_AddressBase [45/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207CB.h.

5.1.2.2159 CLK_AddressBase [46/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207K6.h.

5.1.2.2160 CLK_AddressBase [47/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF62A8.h.

5.1.2.2161 CLK_AddressBase [48/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207K8.h.

5.1.2.2162 CLK_AddressBase [49/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207M8.h.

5.1.2.2163 CLK_AddressBase [50/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207MB.h.

5.1.2.2164 CLK_AddressBase [51/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207R6.h.

5.1.2.2165 CLK_AddressBase [52/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF62A9.h.

5.1.2.2166 CLK_AddressBase [53/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207R8.h.

5.1.2.2167 CLK_AddressBase [54/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207RB.h.

5.1.2.2168 CLK_AddressBase [55/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207S6.h.

5.1.2.2169 CLK_AddressBase [56/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF62AA.h.

5.1.2.2170 CLK_AddressBase [57/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207S8.h.

5.1.2.2171 CLK_AddressBase [58/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S207SB.h.

5.1.2.2172 CLK_AddressBase [59/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF52A6.h.

5.1.2.2173 CLK_AddressBase [60/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S208C6.h.

5.1.2.2174 CLK_AddressBase [61/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF5269.h.

5.1.2.2175 CLK_AddressBase [62/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF628A.h.

5.1.2.2176 CLK_AddressBase [63/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S208C8.h.

5.1.2.2177 CLK_AddressBase [64/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S208CB.h.

5.1.2.2178 CLK_AddressBase [65/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S208R6.h.

5.1.2.2179 CLK_AddressBase [66/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S208R8.h.

5.1.2.2180 CLK_AddressBase [67/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF6388.h.

5.1.2.2181 CLK_AddressBase [68/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8AF5288.h.

5.1.2.2182 CLK_AddressBase [69/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S208S6.h.

5.1.2.2183 CLK_AddressBase [70/70]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 65 of file STM8S007C8.h.

5.1.2.2184 DISABLE_INTERRUPTS

```
#define DISABLE_INTERRUPTS( ) __asm__("sim")
```

disable interrupt handling

Definition at line 169 of file STM8AF_STM8S.h.

5.1.2.2185 DM_AddressBase [1/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8S903F3.h.

5.1.2.2186 DM_AddressBase [2/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8S903K3.h.

5.1.2.2187 DM_AddressBase [3/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8S001J3.h.

5.1.2.2188 DM_AddressBase [4/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8S003F3.h.

5.1.2.2189 DM_AddressBase [5/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8S003K3.h.

5.1.2.2190 DM_AddressBase [6/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8S103F2.h.

5.1.2.2191 DM_AddressBase [7/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8S103F3.h.

5.1.2.2192 DM_AddressBase [8/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8AF6213.h.

5.1.2.2193 DM_AddressBase [9/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8S103K3.h.

5.1.2.2194 DM_AddressBase [10/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8AF6213A.h.

5.1.2.2195 DM_AddressBase [11/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8AF6223.h.

5.1.2.2196 DM_AddressBase [12/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8AF6223A.h.

5.1.2.2197 DM_AddressBase [13/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 76 of file STM8AF6226.h.

5.1.2.2198 DM_AddressBase [14/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 77 of file STM8AF6366.h.

5.1.2.2199 DM_AddressBase [15/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8AF6246.h.

5.1.2.2200 DM_AddressBase [16/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8AF6248.h.

5.1.2.2201 DM_AddressBase [17/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8AF6266.h.

5.1.2.2202 DM_AddressBase [18/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8AF6268.h.

5.1.2.2203 DM_AddressBase [19/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8AF6269.h.

5.1.2.2204 DM_AddressBase [20/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8S005C6.h.

5.1.2.2205 DM_AddressBase [21/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8S005K6.h.

5.1.2.2206 DM_AddressBase [22/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8S105C4.h.

5.1.2.2207 DM_AddressBase [23/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8S105C6.h.

5.1.2.2208 DM_AddressBase [24/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8S105K4.h.

5.1.2.2209 DM_AddressBase [25/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8S105S6.h.

5.1.2.2210 DM_AddressBase [26/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8S105S4.h.

5.1.2.2211 DM_AddressBase [27/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 78 of file STM8S105K6.h.

5.1.2.2212 DM_AddressBase [28/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S007C8.h.

5.1.2.2213 DM_AddressBase [29/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8AF6286.h.

5.1.2.2214 DM_AddressBase [30/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8AF6288.h.

5.1.2.2215 DM_AddressBase [31/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8AF628A.h.

5.1.2.2216 DM_AddressBase [32/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207C6.h.

5.1.2.2217 DM_AddressBase [33/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8AF62A6.h.

5.1.2.2218 DM_AddressBase [34/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207C8.h.

5.1.2.2219 DM_AddressBase [35/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207CB.h.

5.1.2.2220 DM_AddressBase [36/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207K6.h.

5.1.2.2221 DM_AddressBase [37/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207K8.h.

5.1.2.2222 DM_AddressBase [38/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8AF62A8.h.

5.1.2.2223 DM_AddressBase [39/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207M8.h.

5.1.2.2224 DM_AddressBase [40/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207MB.h.

5.1.2.2225 DM_AddressBase [41/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207R6.h.

5.1.2.2226 DM_AddressBase [42/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8AF62A9.h.

5.1.2.2227 DM_AddressBase [43/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207R8.h.

5.1.2.2228 DM_AddressBase [44/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207RB.h.

5.1.2.2229 DM_AddressBase [45/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207S6.h.

5.1.2.2230 DM_AddressBase [46/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207S8.h.

5.1.2.2231 DM_AddressBase [47/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8S207SB.h.

5.1.2.2232 DM_AddressBase [48/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8AF62AA.h.

5.1.2.2233 DM_AddressBase [49/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8AF6289.h.

5.1.2.2234 DM_AddressBase [50/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 81 of file STM8AF6388.h.

5.1.2.2235 DM_AddressBase [51/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8S208S8.h.

5.1.2.2236 DM_AddressBase [52/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8AF5269.h.

5.1.2.2237 DM_AddressBase [53/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8AF52A8.h.

5.1.2.2238 DM_AddressBase [54/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8AF5288.h.

5.1.2.2239 DM_AddressBase [55/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8S208SB.h.

5.1.2.2240 DM_AddressBase [56/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8AF5268.h.

5.1.2.2241 DM_AddressBase [57/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8AF52A9.h.

5.1.2.2242 DM_AddressBase [58/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8S208C8.h.

5.1.2.2243 DM_AddressBase [59/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8S208RB.h.

5.1.2.2244 DM_AddressBase [60/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8AF528A.h.

5.1.2.2245 DM_AddressBase [61/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8AF5289.h.

5.1.2.2246 DM_AddressBase [62/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8AF52AA.h.

5.1.2.2247 DM_AddressBase [63/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8AF5286.h.

5.1.2.2248 DM_AddressBase [64/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8S208C6.h.

5.1.2.2249 DM_AddressBase [65/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8S208CB.h.

5.1.2.2250 DM_AddressBase [66/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8S208MB.h.

5.1.2.2251 DM_AddressBase [67/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8S208R6.h.

5.1.2.2252 DM_AddressBase [68/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8S208R8.h.

5.1.2.2253 DM_AddressBase [69/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8AF52A6.h.

5.1.2.2254 DM_AddressBase [70/70]

```
#define DM_AddressBase 0x7F90
```

Definition at line 82 of file STM8S208S6.h.

5.1.2.2255 ENABLE_INTERRUPTS

```
#define ENABLE_INTERRUPTS( ) __asm__("rim")
```

enable interrupt handling

Definition at line 170 of file STM8AF_STM8S.h.

5.1.2.2256 ENTER_HALT

```
#define ENTER_HALT( ) __asm__("halt")
```

put controller to HALT mode

Definition at line 173 of file STM8AF_STM8S.h.

5.1.2.2257 EXTI_AddressBase [1/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8S903F3.h.

5.1.2.2258 EXTI_AddressBase [2/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8S903K3.h.

5.1.2.2259 EXTI_AddressBase [3/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8AF6226.h.

5.1.2.2260 EXTI_AddressBase [4/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8S001J3.h.

5.1.2.2261 EXTI_AddressBase [5/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8S003F3.h.

5.1.2.2262 EXTI_AddressBase [6/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8S003K3.h.

5.1.2.2263 EXTI_AddressBase [7/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8AF6213.h.

5.1.2.2264 EXTI_AddressBase [8/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8S103F2.h.

5.1.2.2265 EXTI_AddressBase [9/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8S103F3.h.

5.1.2.2266 EXTI_AddressBase [10/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8S103K3.h.

5.1.2.2267 EXTI_AddressBase [11/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8AF6223.h.

5.1.2.2268 EXTI_AddressBase [12/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8AF6223A.h.

5.1.2.2269 EXTI_AddressBase [13/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8AF6213A.h.

5.1.2.2270 EXTI_AddressBase [14/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 60 of file STM8AF6366.h.

5.1.2.2271 EXTI_AddressBase [15/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8AF6248.h.

5.1.2.2272 EXTI_AddressBase [16/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8AF6266.h.

5.1.2.2273 EXTI_AddressBase [17/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8AF6246.h.

5.1.2.2274 EXTI_AddressBase [18/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8AF6268.h.

5.1.2.2275 EXTI_AddressBase [19/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8AF6269.h.

5.1.2.2276 EXTI_AddressBase [20/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8S005C6.h.

5.1.2.2277 EXTI_AddressBase [21/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8S005K6.h.

5.1.2.2278 EXTI_AddressBase [22/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8S105C4.h.

5.1.2.2279 EXTI_AddressBase [23/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8S105C6.h.

5.1.2.2280 EXTI_AddressBase [24/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8S105K4.h.

5.1.2.2281 EXTI_AddressBase [25/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8S105K6.h.

5.1.2.2282 EXTI_AddressBase [26/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8S105S6.h.

5.1.2.2283 EXTI_AddressBase [27/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 61 of file STM8S105S4.h.

5.1.2.2284 EXTI_AddressBase [28/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S208SB.h.

5.1.2.2285 EXTI_AddressBase [29/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF52A8.h.

5.1.2.2286 EXTI_AddressBase [30/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207RB.h.

5.1.2.2287 EXTI_AddressBase [31/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207S8.h.

5.1.2.2288 EXTI_AddressBase [32/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF52A9.h.

5.1.2.2289 EXTI_AddressBase [33/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S208C6.h.

5.1.2.2290 EXTI_AddressBase [34/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S208S8.h.

5.1.2.2291 EXTI_AddressBase [35/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S208R8.h.

5.1.2.2292 EXTI_AddressBase [36/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF5289.h.

5.1.2.2293 EXTI_AddressBase [37/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207C8.h.

5.1.2.2294 EXTI_AddressBase [38/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF52AA.h.

5.1.2.2295 EXTI_AddressBase [39/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF6286.h.

5.1.2.2296 EXTI_AddressBase [40/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207M8.h.

5.1.2.2297 EXTI_AddressBase [41/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207K6.h.

5.1.2.2298 EXTI_AddressBase [42/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207R6.h.

5.1.2.2299 EXTI_AddressBase [43/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF5268.h.

5.1.2.2300 EXTI_AddressBase [44/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF5286.h.

5.1.2.2301 EXTI_AddressBase [45/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF6288.h.

5.1.2.2302 EXTI_AddressBase [46/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S007C8.h.

5.1.2.2303 EXTI_AddressBase [47/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF528A.h.

5.1.2.2304 EXTI_AddressBase [48/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF62A6.h.

5.1.2.2305 EXTI_AddressBase [49/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207C6.h.

5.1.2.2306 EXTI_AddressBase [50/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF5269.h.

5.1.2.2307 EXTI_AddressBase [51/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207CB.h.

5.1.2.2308 EXTI_AddressBase [52/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF62A8.h.

5.1.2.2309 EXTI_AddressBase [53/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207K8.h.

5.1.2.2310 EXTI_AddressBase [54/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207MB.h.

5.1.2.2311 EXTI_AddressBase [55/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF62A9.h.

5.1.2.2312 EXTI_AddressBase [56/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207R8.h.

5.1.2.2313 EXTI_AddressBase [57/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207S6.h.

5.1.2.2314 EXTI_AddressBase [58/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF62AA.h.

5.1.2.2315 EXTI_AddressBase [59/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF52A6.h.

5.1.2.2316 EXTI_AddressBase [60/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S207SB.h.

5.1.2.2317 EXTI_AddressBase [61/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF628A.h.

5.1.2.2318 EXTI_AddressBase [62/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S208C8.h.

5.1.2.2319 EXTI_AddressBase [63/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S208CB.h.

5.1.2.2320 EXTI_AddressBase [64/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S208MB.h.

5.1.2.2321 EXTI_AddressBase [65/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S208R6.h.

5.1.2.2322 EXTI_AddressBase [66/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF5288.h.

5.1.2.2323 EXTI_AddressBase [67/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF6388.h.

5.1.2.2324 EXTI_AddressBase [68/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S208RB.h.

5.1.2.2325 EXTI_AddressBase [69/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8AF6289.h.

5.1.2.2326 EXTI_AddressBase [70/70]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 63 of file STM8S208S6.h.

5.1.2.2327 FLASH_AddressBase [1/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8S903K3.h.

5.1.2.2328 FLASH_AddressBase [2/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8S903F3.h.

5.1.2.2329 FLASH_AddressBase [3/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8AF6366.h.

5.1.2.2330 FLASH_AddressBase [4/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8AF6226.h.

5.1.2.2331 FLASH_AddressBase [5/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8S001J3.h.

5.1.2.2332 FLASH_AddressBase [6/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8S003F3.h.

5.1.2.2333 FLASH_AddressBase [7/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8S003K3.h.

5.1.2.2334 FLASH_AddressBase [8/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8AF6213.h.

5.1.2.2335 FLASH_AddressBase [9/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8S103K3.h.

5.1.2.2336 FLASH_AddressBase [10/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8AF6213A.h.

5.1.2.2337 FLASH_AddressBase [11/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8AF6223.h.

5.1.2.2338 FLASH_AddressBase [12/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8AF6223A.h.

5.1.2.2339 FLASH_AddressBase [13/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8S103F3.h.

5.1.2.2340 FLASH_AddressBase [14/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 59 of file STM8S103F2.h.

5.1.2.2341 FLASH_AddressBase [15/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8AF6246.h.

5.1.2.2342 FLASH_AddressBase [16/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8AF6248.h.

5.1.2.2343 FLASH_AddressBase [17/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8AF6266.h.

5.1.2.2344 FLASH_AddressBase [18/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8AF6268.h.

5.1.2.2345 FLASH_AddressBase [19/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8AF6269.h.

5.1.2.2346 FLASH_AddressBase [20/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8S005C6.h.

5.1.2.2347 FLASH_AddressBase [21/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8S005K6.h.

5.1.2.2348 FLASH_AddressBase [22/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8S105C4.h.

5.1.2.2349 FLASH_AddressBase [23/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8S105K4.h.

5.1.2.2350 FLASH_AddressBase [24/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8S105S4.h.

5.1.2.2351 FLASH_AddressBase [25/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8S105S6.h.

5.1.2.2352 FLASH_AddressBase [26/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8S105K6.h.

5.1.2.2353 FLASH_AddressBase [27/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 60 of file STM8S105C6.h.

5.1.2.2354 FLASH_AddressBase [28/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S208S8.h.

5.1.2.2355 FLASH_AddressBase [29/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF52A8.h.

5.1.2.2356 FLASH_AddressBase [30/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF52A6.h.

5.1.2.2357 FLASH_AddressBase [31/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF52A9.h.

5.1.2.2358 FLASH_AddressBase [32/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S208SB.h.

5.1.2.2359 FLASH_AddressBase [33/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF5268.h.

5.1.2.2360 FLASH_AddressBase [34/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S208CB.h.

5.1.2.2361 FLASH_AddressBase [35/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S208R8.h.

5.1.2.2362 FLASH_AddressBase [36/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S208C8.h.

5.1.2.2363 FLASH_AddressBase [37/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S208S6.h.

5.1.2.2364 FLASH_AddressBase [38/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF5288.h.

5.1.2.2365 FLASH_AddressBase [39/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF52AA.h.

5.1.2.2366 FLASH_AddressBase [40/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S208MB.h.

5.1.2.2367 FLASH_AddressBase [41/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF6286.h.

5.1.2.2368 FLASH_AddressBase [42/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF5286.h.

5.1.2.2369 FLASH_AddressBase [43/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF6288.h.

5.1.2.2370 FLASH_AddressBase [44/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S007C8.h.

5.1.2.2371 FLASH_AddressBase [45/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF6289.h.

5.1.2.2372 FLASH_AddressBase [46/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF628A.h.

5.1.2.2373 FLASH_AddressBase [47/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF528A.h.

5.1.2.2374 FLASH_AddressBase [48/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF5289.h.

5.1.2.2375 FLASH_AddressBase [49/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF62A6.h.

5.1.2.2376 FLASH_AddressBase [50/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207C6.h.

5.1.2.2377 FLASH_AddressBase [51/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF5269.h.

5.1.2.2378 FLASH_AddressBase [52/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207C8.h.

5.1.2.2379 FLASH_AddressBase [53/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207CB.h.

5.1.2.2380 FLASH_AddressBase [54/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF62A8.h.

5.1.2.2381 FLASH_AddressBase [55/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207K6.h.

5.1.2.2382 FLASH_AddressBase [56/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207K8.h.

5.1.2.2383 FLASH_AddressBase [57/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207M8.h.

5.1.2.2384 FLASH_AddressBase [58/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207MB.h.

5.1.2.2385 FLASH_AddressBase [59/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF62A9.h.

5.1.2.2386 FLASH_AddressBase [60/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207R6.h.

5.1.2.2387 FLASH_AddressBase [61/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207R8.h.

5.1.2.2388 FLASH_AddressBase [62/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207RB.h.

5.1.2.2389 FLASH_AddressBase [63/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207S6.h.

5.1.2.2390 FLASH_AddressBase [64/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF62AA.h.

5.1.2.2391 FLASH_AddressBase [65/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207S8.h.

5.1.2.2392 FLASH_AddressBase [66/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S207SB.h.

5.1.2.2393 FLASH_AddressBase [67/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S208C6.h.

5.1.2.2394 FLASH_AddressBase [68/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S208R6.h.

5.1.2.2395 FLASH_AddressBase [69/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8AF6388.h.

5.1.2.2396 FLASH_AddressBase [70/70]

```
#define FLASH_AddressBase 0x505A
```

Definition at line 62 of file STM8S208RB.h.

5.1.2.2397 I2C_AddressBase [1/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8S903F3.h.

5.1.2.2398 I2C_AddressBase [2/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8S903K3.h.

5.1.2.2399 I2C_AddressBase [3/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8AF6226.h.

5.1.2.2400 I2C_AddressBase [4/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8S001J3.h.

5.1.2.2401 I2C_AddressBase [5/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8S003F3.h.

5.1.2.2402 I2C_AddressBase [6/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8S003K3.h.

5.1.2.2403 I2C_AddressBase [7/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8AF6213.h.

5.1.2.2404 I2C_AddressBase [8/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8S103F2.h.

5.1.2.2405 I2C_AddressBase [9/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8S103F3.h.

5.1.2.2406 I2C_AddressBase [10/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8S103K3.h.

5.1.2.2407 I2C_AddressBase [11/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8AF6213A.h.

5.1.2.2408 I2C_AddressBase [12/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8AF6223.h.

5.1.2.2409 I2C_AddressBase [13/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8AF6223A.h.

5.1.2.2410 I2C_AddressBase [14/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 68 of file STM8AF6366.h.

5.1.2.2411 I2C_AddressBase [15/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8AF6246.h.

5.1.2.2412 I2C_AddressBase [16/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8AF6248.h.

5.1.2.2413 I2C_AddressBase [17/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8AF6266.h.

5.1.2.2414 I2C_AddressBase [18/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8AF6268.h.

5.1.2.2415 I2C_AddressBase [19/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8AF6269.h.

5.1.2.2416 I2C_AddressBase [20/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8S005C6.h.

5.1.2.2417 I2C_AddressBase [21/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8S005K6.h.

5.1.2.2418 I2C_AddressBase [22/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8S105C4.h.

5.1.2.2419 I2C_AddressBase [23/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8S105C6.h.

5.1.2.2420 I2C_AddressBase [24/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8S105K4.h.

5.1.2.2421 I2C_AddressBase [25/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8S105K6.h.

5.1.2.2422 I2C_AddressBase [26/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8S105S4.h.

5.1.2.2423 I2C_AddressBase [27/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 69 of file STM8S105S6.h.

5.1.2.2424 I2C_AddressBase [28/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF6289.h.

5.1.2.2425 I2C_AddressBase [29/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207RB.h.

5.1.2.2426 I2C_AddressBase [30/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF5288.h.

5.1.2.2427 I2C_AddressBase [31/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207R6.h.

5.1.2.2428 I2C_AddressBase [32/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF52A8.h.

5.1.2.2429 I2C_AddressBase [33/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207S8.h.

5.1.2.2430 I2C_AddressBase [34/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S208SB.h.

5.1.2.2431 I2C_AddressBase [35/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S208S8.h.

5.1.2.2432 I2C_AddressBase [36/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF52A9.h.

5.1.2.2433 I2C_AddressBase [37/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S208C6.h.

5.1.2.2434 I2C_AddressBase [38/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S208R8.h.

5.1.2.2435 I2C_AddressBase [39/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF52A6.h.

5.1.2.2436 I2C_AddressBase [40/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207C8.h.

5.1.2.2437 I2C_AddressBase [41/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207M8.h.

5.1.2.2438 I2C_AddressBase [42/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF5289.h.

5.1.2.2439 I2C_AddressBase [43/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF5268.h.

5.1.2.2440 I2C_AddressBase [44/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF6286.h.

5.1.2.2441 I2C_AddressBase [45/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF52AA.h.

5.1.2.2442 I2C_AddressBase [46/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207K6.h.

5.1.2.2443 I2C_AddressBase [47/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF6288.h.

5.1.2.2444 I2C_AddressBase [48/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF628A.h.

5.1.2.2445 I2C_AddressBase [49/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF528A.h.

5.1.2.2446 I2C_AddressBase [50/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF62A6.h.

5.1.2.2447 I2C_AddressBase [51/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207C6.h.

5.1.2.2448 I2C_AddressBase [52/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207CB.h.

5.1.2.2449 I2C_AddressBase [53/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF62A8.h.

5.1.2.2450 I2C_AddressBase [54/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207K8.h.

5.1.2.2451 I2C_AddressBase [55/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207MB.h.

5.1.2.2452 I2C_AddressBase [56/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF62A9.h.

5.1.2.2453 I2C_AddressBase [57/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207R8.h.

5.1.2.2454 I2C_AddressBase [58/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207S6.h.

5.1.2.2455 I2C_AddressBase [59/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF62AA.h.

5.1.2.2456 I2C_AddressBase [60/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S207SB.h.

5.1.2.2457 I2C_AddressBase [61/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S208C8.h.

5.1.2.2458 I2C_AddressBase [62/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S208CB.h.

5.1.2.2459 I2C_AddressBase [63/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S208MB.h.

5.1.2.2460 I2C_AddressBase [64/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF5286.h.

5.1.2.2461 I2C_AddressBase [65/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S007C8.h.

5.1.2.2462 I2C_AddressBase [66/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S208R6.h.

5.1.2.2463 I2C_AddressBase [67/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF5269.h.

5.1.2.2464 I2C_AddressBase [68/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8AF6388.h.

5.1.2.2465 I2C_AddressBase [69/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S208RB.h.

5.1.2.2466 I2C_AddressBase [70/70]

```
#define I2C_AddressBase 0x5210
```

Definition at line 71 of file STM8S208S6.h.

5.1.2.2467 ISR_HANDLER

```
#define ISR_HANDLER(
              func,
              irq ) void func(void) __interrupt(irq)
```

handler for interrupt service routine

Definition at line 160 of file STM8AF_STM8S.h.

5.1.2.2468 ISR_HANDLER_TRAP

```
#define ISR_HANDLER_TRAP(
              func ) void func() __trap
```

handler for trap service routine

Definition at line 162 of file STM8AF_STM8S.h.

5.1.2.2469 ITC_AddressBase [1/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8S103F2.h.

5.1.2.2470 ITC_AddressBase [2/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8S903K3.h.

5.1.2.2471 ITC_AddressBase [3/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8S903F3.h.

5.1.2.2472 ITC_AddressBase [4/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8S001J3.h.

5.1.2.2473 ITC_AddressBase [5/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8S003F3.h.

5.1.2.2474 ITC_AddressBase [6/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8S003K3.h.

5.1.2.2475 ITC_AddressBase [7/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8AF6213.h.

5.1.2.2476 ITC_AddressBase [8/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8AF6223.h.

5.1.2.2477 ITC_AddressBase [9/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8AF6213A.h.

5.1.2.2478 ITC_AddressBase [10/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8AF6223A.h.

5.1.2.2479 ITC_AddressBase [11/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8S103F3.h.

5.1.2.2480 ITC_AddressBase [12/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8S103K3.h.

5.1.2.2481 ITC_AddressBase [13/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 75 of file STM8AF6226.h.

5.1.2.2482 ITC_AddressBase [14/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 76 of file STM8AF6366.h.

5.1.2.2483 ITC_AddressBase [15/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8S005C6.h.

5.1.2.2484 ITC_AddressBase [16/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8AF6246.h.

5.1.2.2485 ITC_AddressBase [17/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8S005K6.h.

5.1.2.2486 ITC_AddressBase [18/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8AF6248.h.

5.1.2.2487 ITC_AddressBase [19/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8AF6266.h.

5.1.2.2488 ITC_AddressBase [20/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8AF6268.h.

5.1.2.2489 ITC_AddressBase [21/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8AF6269.h.

5.1.2.2490 ITC_AddressBase [22/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8S105S4.h.

5.1.2.2491 ITC_AddressBase [23/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8S105C4.h.

5.1.2.2492 ITC_AddressBase [24/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8S105K6.h.

5.1.2.2493 ITC_AddressBase [25/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8S105C6.h.

5.1.2.2494 ITC_AddressBase [26/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8S105K4.h.

5.1.2.2495 ITC_AddressBase [27/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 77 of file STM8S105S6.h.

5.1.2.2496 ITC_AddressBase [28/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S007C8.h.

5.1.2.2497 ITC_AddressBase [29/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8AF6288.h.

5.1.2.2498 ITC_AddressBase [30/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207R8.h.

5.1.2.2499 ITC_AddressBase [31/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207RB.h.

5.1.2.2500 ITC_AddressBase [32/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207S6.h.

5.1.2.2501 ITC_AddressBase [33/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207S8.h.

5.1.2.2502 ITC_AddressBase [34/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207SB.h.

5.1.2.2503 ITC_AddressBase [35/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8AF62AA.h.

5.1.2.2504 ITC_AddressBase [36/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207CB.h.

5.1.2.2505 ITC_AddressBase [37/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207C8.h.

5.1.2.2506 ITC_AddressBase [38/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207M8.h.

5.1.2.2507 ITC_AddressBase [39/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207K8.h.

5.1.2.2508 ITC_AddressBase [40/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8AF6286.h.

5.1.2.2509 ITC_AddressBase [41/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207K6.h.

5.1.2.2510 ITC_AddressBase [42/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207R6.h.

5.1.2.2511 ITC_AddressBase [43/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207MB.h.

5.1.2.2512 ITC_AddressBase [44/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8AF6289.h.

5.1.2.2513 ITC_AddressBase [45/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8AF628A.h.

5.1.2.2514 ITC_AddressBase [46/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8AF62A6.h.

5.1.2.2515 ITC_AddressBase [47/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8AF62A8.h.

5.1.2.2516 ITC_AddressBase [48/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8AF62A9.h.

5.1.2.2517 ITC_AddressBase [49/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8S207C6.h.

5.1.2.2518 ITC_AddressBase [50/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 80 of file STM8AF6388.h.

5.1.2.2519 ITC_AddressBase [51/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8AF5268.h.

5.1.2.2520 ITC_AddressBase [52/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8AF5269.h.

5.1.2.2521 ITC_AddressBase [53/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8AF5288.h.

5.1.2.2522 ITC_AddressBase [54/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8AF52A8.h.

5.1.2.2523 ITC_AddressBase [55/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8S208SB.h.

5.1.2.2524 ITC_AddressBase [56/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8S208C8.h.

5.1.2.2525 ITC_AddressBase [57/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8S208S8.h.

5.1.2.2526 ITC_AddressBase [58/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8S208R8.h.

5.1.2.2527 ITC_AddressBase [59/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8AF52A6.h.

5.1.2.2528 ITC_AddressBase [60/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8S208R6.h.

5.1.2.2529 ITC_AddressBase [61/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8AF52A9.h.

5.1.2.2530 ITC_AddressBase [62/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8S208CB.h.

5.1.2.2531 ITC_AddressBase [63/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8S208MB.h.

5.1.2.2532 ITC_AddressBase [64/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8S208C6.h.

5.1.2.2533 ITC_AddressBase [65/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8S208S6.h.

5.1.2.2534 ITC_AddressBase [66/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8S208RB.h.

5.1.2.2535 ITC_AddressBase [67/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8AF5289.h.

5.1.2.2536 ITC_AddressBase [68/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8AF52AA.h.

5.1.2.2537 ITC_AddressBase [69/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8AF5286.h.

5.1.2.2538 ITC_AddressBase [70/70]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 81 of file STM8AF528A.h.

5.1.2.2539 IWDG_AddressBase [1/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8S103K3.h.

5.1.2.2540 IWDG_AddressBase [2/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8S103F2.h.

5.1.2.2541 IWDG_AddressBase [3/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8S903F3.h.

5.1.2.2542 IWDG_AddressBase [4/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8AF6213.h.

5.1.2.2543 IWDG_AddressBase [5/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8S903K3.h.

5.1.2.2544 IWDG_AddressBase [6/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8S001J3.h.

5.1.2.2545 IWDG_AddressBase [7/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8S003F3.h.

5.1.2.2546 IWDG_AddressBase [8/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8S003K3.h.

5.1.2.2547 IWDG_AddressBase [9/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8AF6213A.h.

5.1.2.2548 IWDG_AddressBase [10/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8AF6223.h.

5.1.2.2549 IWDG_AddressBase [11/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8AF6223A.h.

5.1.2.2550 IWDG_AddressBase [12/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8AF6366.h.

5.1.2.2551 IWDG_AddressBase [13/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8AF6226.h.

5.1.2.2552 IWDG_AddressBase [14/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 64 of file STM8S103F3.h.

5.1.2.2553 IWDG_AddressBase [15/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8AF6246.h.

5.1.2.2554 IWDG_AddressBase [16/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8S005K6.h.

5.1.2.2555 IWDG_AddressBase [17/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8S005C6.h.

5.1.2.2556 IWDG_AddressBase [18/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8AF6248.h.

5.1.2.2557 IWDG_AddressBase [19/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8AF6266.h.

5.1.2.2558 IWDG_AddressBase [20/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8AF6268.h.

5.1.2.2559 IWDG_AddressBase [21/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8AF6269.h.

5.1.2.2560 IWDG_AddressBase [22/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8S105K6.h.

5.1.2.2561 IWDG_AddressBase [23/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8S105S6.h.

5.1.2.2562 IWDG_AddressBase [24/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8S105S4.h.

5.1.2.2563 IWDG_AddressBase [25/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8S105C4.h.

5.1.2.2564 IWDG_AddressBase [26/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8S105C6.h.

5.1.2.2565 IWDG_AddressBase [27/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 65 of file STM8S105K4.h.

5.1.2.2566 IWDG_AddressBase [28/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF5268.h.

5.1.2.2567 IWDG_AddressBase [29/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF5288.h.

5.1.2.2568 IWDG_AddressBase [30/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207RB.h.

5.1.2.2569 IWDG_AddressBase [31/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF52A8.h.

5.1.2.2570 IWDG_AddressBase [32/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207R8.h.

5.1.2.2571 IWDG_AddressBase [33/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF6288.h.

5.1.2.2572 IWDG_AddressBase [34/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207S6.h.

5.1.2.2573 IWDG_AddressBase [35/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF5269.h.

5.1.2.2574 IWDG_AddressBase [36/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF62AA.h.

5.1.2.2575 IWDG_AddressBase [37/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207S8.h.

5.1.2.2576 IWDG_AddressBase [38/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S208SB.h.

5.1.2.2577 IWDG_AddressBase [39/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207SB.h.

5.1.2.2578 IWDG_AddressBase [40/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF52A9.h.

5.1.2.2579 IWDG_AddressBase [41/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S208S8.h.

5.1.2.2580 IWDG_AddressBase [42/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S208CB.h.

5.1.2.2581 IWDG_AddressBase [43/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S208C6.h.

5.1.2.2582 IWDG_AddressBase [44/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S208R8.h.

5.1.2.2583 IWDG_AddressBase [45/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S208R6.h.

5.1.2.2584 IWDG_AddressBase [46/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S208C8.h.

5.1.2.2585 IWDG_AddressBase [47/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S208RB.h.

5.1.2.2586 IWDG_AddressBase [48/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF5289.h.

5.1.2.2587 IWDG_AddressBase [49/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207C8.h.

5.1.2.2588 IWDG_AddressBase [50/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207CB.h.

5.1.2.2589 IWDG_AddressBase [51/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207M8.h.

5.1.2.2590 IWDG_AddressBase [52/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF6286.h.

5.1.2.2591 IWDG_AddressBase [53/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF52AA.h.

5.1.2.2592 IWDG_AddressBase [54/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207K6.h.

5.1.2.2593 IWDG_AddressBase [55/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207K8.h.

5.1.2.2594 IWDG_AddressBase [56/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207MB.h.

5.1.2.2595 IWDG_AddressBase [57/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF62A8.h.

5.1.2.2596 IWDG_AddressBase [58/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF62A9.h.

5.1.2.2597 IWDG_AddressBase [59/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207R6.h.

5.1.2.2598 IWDG_AddressBase [60/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF5286.h.

5.1.2.2599 IWDG_AddressBase [61/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF6289.h.

5.1.2.2600 IWDG_AddressBase [62/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF528A.h.

5.1.2.2601 IWDG_AddressBase [63/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF62A6.h.

5.1.2.2602 IWDG_AddressBase [64/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S207C6.h.

5.1.2.2603 IWDG_AddressBase [65/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF52A6.h.

5.1.2.2604 IWDG_AddressBase [66/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF628A.h.

5.1.2.2605 IWDG_AddressBase [67/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S208MB.h.

5.1.2.2606 IWDG_AddressBase [68/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8AF6388.h.

5.1.2.2607 IWDG_AddressBase [69/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S007C8.h.

5.1.2.2608 IWDG_AddressBase [70/70]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 67 of file STM8S208S6.h.

5.1.2.2609 NOP

```
#define NOP( ) __asm__("nop")
```

perform a nop() operation (=minimum delay)

Definition at line 168 of file STM8AF_STM8S.h.

5.1.2.2610 OPT_AddressBase [1/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S208S8.h.

5.1.2.2611 OPT_AddressBase [2/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S208SB.h.

5.1.2.2612 OPT_AddressBase [3/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6246.h.

5.1.2.2613 OPT_AddressBase [4/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207RB.h.

5.1.2.2614 OPT_AddressBase [5/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6289.h.

5.1.2.2615 OPT_AddressBase [6/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6248.h.

5.1.2.2616 OPT_AddressBase [7/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6213.h.

5.1.2.2617 OPT_AddressBase [8/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6223A.h.

5.1.2.2618 OPT_AddressBase [9/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF5268.h.

5.1.2.2619 OPT_AddressBase [10/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6266.h.

5.1.2.2620 OPT_AddressBase [11/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207S8.h.

5.1.2.2621 OPT_AddressBase [12/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF52A9.h.

5.1.2.2622 OPT_AddressBase [13/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF5289.h.

5.1.2.2623 OPT_AddressBase [14/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207CB.h.

5.1.2.2624 OPT_AddressBase [15/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207S6.h.

5.1.2.2625 OPT_AddressBase [16/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S903F3.h.

5.1.2.2626 OPT_AddressBase [17/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207SB.h.

5.1.2.2627 OPT_AddressBase [18/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF5286.h.

5.1.2.2628 OPT_AddressBase [19/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6268.h.

5.1.2.2629 OPT_AddressBase [20/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S903K3.h.

5.1.2.2630 OPT_AddressBase [21/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF52A8.h.

5.1.2.2631 OPT_AddressBase [22/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S208C6.h.

5.1.2.2632 OPT_AddressBase [23/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S208R8.h.

5.1.2.2633 OPT_AddressBase [24/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6269.h.

5.1.2.2634 OPT_AddressBase [25/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207C8.h.

5.1.2.2635 OPT_AddressBase [26/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF5269.h.

5.1.2.2636 OPT_AddressBase [27/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF52AA.h.

5.1.2.2637 OPT_AddressBase [28/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6286.h.

5.1.2.2638 OPT_AddressBase [29/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S208R6.h.

5.1.2.2639 OPT_AddressBase [30/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207K6.h.

5.1.2.2640 OPT_AddressBase [31/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207M8.h.

5.1.2.2641 OPT_AddressBase [32/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207K8.h.

5.1.2.2642 OPT_AddressBase [33/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6288.h.

5.1.2.2643 OPT_AddressBase [34/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207R6.h.

5.1.2.2644 OPT_AddressBase [35/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S001J3.h.

5.1.2.2645 OPT_AddressBase [36/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF5288.h.

5.1.2.2646 OPT_AddressBase [37/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207R8.h.

5.1.2.2647 OPT_AddressBase [38/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S003F3.h.

5.1.2.2648 OPT_AddressBase [39/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S003K3.h.

5.1.2.2649 OPT_AddressBase [40/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207MB.h.

5.1.2.2650 OPT_AddressBase [41/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S005C6.h.

5.1.2.2651 OPT_AddressBase [42/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF528A.h.

5.1.2.2652 OPT_AddressBase [43/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S005K6.h.

5.1.2.2653 OPT_AddressBase [44/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S208S6.h.

5.1.2.2654 OPT_AddressBase [45/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S007C8.h.

5.1.2.2655 OPT_AddressBase [46/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S103F2.h.

5.1.2.2656 OPT_AddressBase [47/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S103F3.h.

5.1.2.2657 OPT_AddressBase [48/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S103K3.h.

5.1.2.2658 OPT_AddressBase [49/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S105C4.h.

5.1.2.2659 OPT_AddressBase [50/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6213A.h.

5.1.2.2660 OPT_AddressBase [51/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S105K4.h.

5.1.2.2661 OPT_AddressBase [52/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF62A6.h.

5.1.2.2662 OPT_AddressBase [53/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S105S4.h.

5.1.2.2663 OPT_AddressBase [54/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF62A8.h.

5.1.2.2664 OPT_AddressBase [55/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6223.h.

5.1.2.2665 OPT_AddressBase [56/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF52A6.h.

5.1.2.2666 OPT_AddressBase [57/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF62A9.h.

5.1.2.2667 OPT_AddressBase [58/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S105S6.h.

5.1.2.2668 OPT_AddressBase [59/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF62AA.h.

5.1.2.2669 OPT_AddressBase [60/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S207C6.h.

5.1.2.2670 OPT_AddressBase [61/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6366.h.

5.1.2.2671 OPT_AddressBase [62/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S208C8.h.

5.1.2.2672 OPT_AddressBase [63/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S105K6.h.

5.1.2.2673 OPT_AddressBase [64/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6226.h.

5.1.2.2674 OPT_AddressBase [65/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S208CB.h.

5.1.2.2675 OPT_AddressBase [66/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S208MB.h.

5.1.2.2676 OPT_AddressBase [67/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF6388.h.

5.1.2.2677 OPT_AddressBase [68/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S105C6.h.

5.1.2.2678 OPT_AddressBase [69/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8S208RB.h.

5.1.2.2679 OPT_AddressBase [70/70]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8AF628A.h.

5.1.2.2680 PORTA_AddressBase [1/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S208S8.h.

5.1.2.2681 PORTA_AddressBase [2/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S103K3.h.

5.1.2.2682 PORTA_AddressBase [3/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF52A8.h.

5.1.2.2683 PORTA_AddressBase [4/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6246.h.

5.1.2.2684 PORTA_AddressBase [5/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6213.h.

5.1.2.2685 PORTA_AddressBase [6/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207RB.h.

5.1.2.2686 PORTA_AddressBase [7/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF528A.h.

5.1.2.2687 PORTA_AddressBase [8/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6289.h.

5.1.2.2688 PORTA_AddressBase [9/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6248.h.

5.1.2.2689 PORTA_AddressBase [10/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6223A.h.

5.1.2.2690 PORTA_AddressBase [11/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF5288.h.

5.1.2.2691 PORTA_AddressBase [12/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6266.h.

5.1.2.2692 PORTA_AddressBase [13/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF52A9.h.

5.1.2.2693 PORTA_AddressBase [14/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6223.h.

5.1.2.2694 PORTA_AddressBase [15/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF5289.h.

5.1.2.2695 PORTA_AddressBase [16/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S903F3.h.

5.1.2.2696 PORTA_AddressBase [17/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6268.h.

5.1.2.2697 PORTA_AddressBase [18/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S903K3.h.

5.1.2.2698 PORTA_AddressBase [19/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S208SB.h.

5.1.2.2699 PORTA_AddressBase [20/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF5286.h.

5.1.2.2700 PORTA_AddressBase [21/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S208R8.h.

5.1.2.2701 PORTA_AddressBase [22/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6269.h.

5.1.2.2702 PORTA_AddressBase [23/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6226.h.

5.1.2.2703 PORTA_AddressBase [24/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF52AA.h.

5.1.2.2704 PORTA_AddressBase [25/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S208RB.h.

5.1.2.2705 PORTA_AddressBase [26/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S208S6.h.

5.1.2.2706 PORTA_AddressBase [27/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S208MB.h.

5.1.2.2707 PORTA_AddressBase [28/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6366.h.

5.1.2.2708 PORTA_AddressBase [29/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207C8.h.

5.1.2.2709 PORTA_AddressBase [30/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6286.h.

5.1.2.2710 PORTA_AddressBase [31/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF5269.h.

5.1.2.2711 PORTA_AddressBase [32/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207CB.h.

5.1.2.2712 PORTA_AddressBase [33/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207M8.h.

5.1.2.2713 PORTA_AddressBase [34/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6288.h.

5.1.2.2714 PORTA_AddressBase [35/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S001J3.h.

5.1.2.2715 PORTA_AddressBase [36/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S003F3.h.

5.1.2.2716 PORTA_AddressBase [37/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S003K3.h.

5.1.2.2717 PORTA_AddressBase [38/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S005C6.h.

5.1.2.2718 PORTA_AddressBase [39/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S005K6.h.

5.1.2.2719 PORTA_AddressBase [40/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S007C8.h.

5.1.2.2720 PORTA_AddressBase [41/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S103F3.h.

5.1.2.2721 PORTA_AddressBase [42/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S105C4.h.

5.1.2.2722 PORTA_AddressBase [43/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6213A.h.

5.1.2.2723 PORTA_AddressBase [44/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S105K4.h.

5.1.2.2724 PORTA_AddressBase [45/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S208CB.h.

5.1.2.2725 PORTA_AddressBase [46/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207C6.h.

5.1.2.2726 PORTA_AddressBase [47/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF5268.h.

5.1.2.2727 PORTA_AddressBase [48/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF62A8.h.

5.1.2.2728 PORTA_AddressBase [49/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207K6.h.

5.1.2.2729 PORTA_AddressBase [50/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207K8.h.

5.1.2.2730 PORTA_AddressBase [51/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S105S4.h.

5.1.2.2731 PORTA_AddressBase [52/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S105S6.h.

5.1.2.2732 PORTA_AddressBase [53/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF62A9.h.

5.1.2.2733 PORTA_AddressBase [54/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207MB.h.

5.1.2.2734 PORTA_AddressBase [55/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF52A6.h.

5.1.2.2735 PORTA_AddressBase [56/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF62A6.h.

5.1.2.2736 PORTA_AddressBase [57/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207R6.h.

5.1.2.2737 PORTA_AddressBase [58/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207R8.h.

5.1.2.2738 PORTA_AddressBase [59/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S105K6.h.

5.1.2.2739 PORTA_AddressBase [60/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF62AA.h.

5.1.2.2740 PORTA_AddressBase [61/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207S6.h.

5.1.2.2741 PORTA_AddressBase [62/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207S8.h.

5.1.2.2742 PORTA_AddressBase [63/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S207SB.h.

5.1.2.2743 PORTA_AddressBase [64/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S208C6.h.

5.1.2.2744 PORTA_AddressBase [65/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S208C8.h.

5.1.2.2745 PORTA_AddressBase [66/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF6388.h.

5.1.2.2746 PORTA_AddressBase [67/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S208R6.h.

5.1.2.2747 PORTA_AddressBase [68/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S103F2.h.

5.1.2.2748 PORTA_AddressBase [69/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8S105C6.h.

5.1.2.2749 PORTA_AddressBase [70/70]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8AF628A.h.

5.1.2.2750 PORTB_AddressBase [1/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S103F2.h.

5.1.2.2751 PORTB_AddressBase [2/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S005C6.h.

5.1.2.2752 PORTB_AddressBase [3/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S005K6.h.

5.1.2.2753 PORTB_AddressBase [4/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF5268.h.

5.1.2.2754 PORTB_AddressBase [5/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6289.h.

5.1.2.2755 PORTB_AddressBase [6/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S903K3.h.

5.1.2.2756 PORTB_AddressBase [7/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207RB.h.

5.1.2.2757 PORTB_AddressBase [8/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6213.h.

5.1.2.2758 PORTB_AddressBase [9/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S007C8.h.

5.1.2.2759 PORTB_AddressBase [10/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6248.h.

5.1.2.2760 PORTB_AddressBase [11/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S001J3.h.

5.1.2.2761 PORTB_AddressBase [12/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6266.h.

5.1.2.2762 PORTB_AddressBase [13/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S003F3.h.

5.1.2.2763 PORTB_AddressBase [14/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF52A9.h.

5.1.2.2764 PORTB_AddressBase [15/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207S8.h.

5.1.2.2765 PORTB_AddressBase [16/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S208SB.h.

5.1.2.2766 PORTB_AddressBase [17/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S903F3.h.

5.1.2.2767 PORTB_AddressBase [18/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207SB.h.

5.1.2.2768 PORTB_AddressBase [19/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF52A8.h.

5.1.2.2769 PORTB_AddressBase [20/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF5289.h.

5.1.2.2770 PORTB_AddressBase [21/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6268.h.

5.1.2.2771 PORTB_AddressBase [22/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6246.h.

5.1.2.2772 PORTB_AddressBase [23/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S208C6.h.

5.1.2.2773 PORTB_AddressBase [24/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S208R8.h.

5.1.2.2774 PORTB_AddressBase [25/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF5286.h.

5.1.2.2775 PORTB_AddressBase [26/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6269.h.

5.1.2.2776 PORTB_AddressBase [27/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6223.h.

5.1.2.2777 PORTB_AddressBase [28/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S208CB.h.

5.1.2.2778 PORTB_AddressBase [29/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S208S8.h.

5.1.2.2779 PORTB_AddressBase [30/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF52AA.h.

5.1.2.2780 PORTB_AddressBase [31/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6286.h.

5.1.2.2781 PORTB_AddressBase [32/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S208R6.h.

5.1.2.2782 PORTB_AddressBase [33/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6366.h.

5.1.2.2783 PORTB_AddressBase [34/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207C8.h.

5.1.2.2784 PORTB_AddressBase [35/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207CB.h.

5.1.2.2785 PORTB_AddressBase [36/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207K6.h.

5.1.2.2786 PORTB_AddressBase [37/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207M8.h.

5.1.2.2787 PORTB_AddressBase [38/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207K8.h.

5.1.2.2788 PORTB_AddressBase [39/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207R6.h.

5.1.2.2789 PORTB_AddressBase [40/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF5269.h.

5.1.2.2790 PORTB_AddressBase [41/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6288.h.

5.1.2.2791 PORTB_AddressBase [42/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF52A6.h.

5.1.2.2792 PORTB_AddressBase [43/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207R8.h.

5.1.2.2793 PORTB_AddressBase [44/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S003K3.h.

5.1.2.2794 PORTB_AddressBase [45/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207MB.h.

5.1.2.2795 PORTB_AddressBase [46/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF528A.h.

5.1.2.2796 PORTB_AddressBase [47/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S103K3.h.

5.1.2.2797 PORTB_AddressBase [48/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF628A.h.

5.1.2.2798 PORTB_AddressBase [49/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6226.h.

5.1.2.2799 PORTB_AddressBase [50/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S208RB.h.

5.1.2.2800 PORTB_AddressBase [51/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S105C6.h.

5.1.2.2801 PORTB_AddressBase [52/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6213A.h.

5.1.2.2802 PORTB_AddressBase [53/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF62A6.h.

5.1.2.2803 PORTB_AddressBase [54/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S105S6.h.

5.1.2.2804 PORTB_AddressBase [55/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF62A8.h.

5.1.2.2805 PORTB_AddressBase [56/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF62A9.h.

5.1.2.2806 PORTB_AddressBase [57/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S105S4.h.

5.1.2.2807 PORTB_AddressBase [58/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6223A.h.

5.1.2.2808 PORTB_AddressBase [59/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF62AA.h.

5.1.2.2809 PORTB_AddressBase [60/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207S6.h.

5.1.2.2810 PORTB_AddressBase [61/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF5288.h.

5.1.2.2811 PORTB_AddressBase [62/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S105K4.h.

5.1.2.2812 PORTB_AddressBase [63/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S105K6.h.

5.1.2.2813 PORTB_AddressBase [64/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S207C6.h.

5.1.2.2814 PORTB_AddressBase [65/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S208C8.h.

5.1.2.2815 PORTB_AddressBase [66/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S208MB.h.

5.1.2.2816 PORTB_AddressBase [67/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8AF6388.h.

5.1.2.2817 PORTB_AddressBase [68/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S105C4.h.

5.1.2.2818 PORTB_AddressBase [69/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S208S6.h.

5.1.2.2819 PORTB_AddressBase [70/70]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8S103F3.h.

5.1.2.2820 PORTC_AddressBase [1/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S103F2.h.

5.1.2.2821 PORTC_AddressBase [2/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S103K3.h.

5.1.2.2822 PORTC_AddressBase [3/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF5269.h.

5.1.2.2823 PORTC_AddressBase [4/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S903F3.h.

5.1.2.2824 PORTC_AddressBase [5/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6213.h.

5.1.2.2825 PORTC_AddressBase [6/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6223A.h.

5.1.2.2826 PORTC_AddressBase [7/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF5268.h.

5.1.2.2827 PORTC_AddressBase [8/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207RB.h.

5.1.2.2828 PORTC_AddressBase [9/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6248.h.

5.1.2.2829 PORTC_AddressBase [10/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6288.h.

5.1.2.2830 PORTC_AddressBase [11/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S001J3.h.

5.1.2.2831 PORTC_AddressBase [12/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207K6.h.

5.1.2.2832 PORTC_AddressBase [13/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S208S8.h.

5.1.2.2833 PORTC_AddressBase [14/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF62AA.h.

5.1.2.2834 PORTC_AddressBase [15/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6266.h.

5.1.2.2835 PORTC_AddressBase [16/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207MB.h.

5.1.2.2836 PORTC_AddressBase [17/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF62A8.h.

5.1.2.2837 PORTC_AddressBase [18/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF52A9.h.

5.1.2.2838 PORTC_AddressBase [19/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF52A8.h.

5.1.2.2839 PORTC_AddressBase [20/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S903K3.h.

5.1.2.2840 PORTC_AddressBase [21/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S208SB.h.

5.1.2.2841 PORTC_AddressBase [22/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF5289.h.

5.1.2.2842 PORTC_AddressBase [23/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6268.h.

5.1.2.2843 PORTC_AddressBase [24/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF5288.h.

5.1.2.2844 PORTC_AddressBase [25/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S208R8.h.

5.1.2.2845 PORTC_AddressBase [26/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S208MB.h.

5.1.2.2846 PORTC_AddressBase [27/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6246.h.

5.1.2.2847 PORTC_AddressBase [28/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S208S6.h.

5.1.2.2848 PORTC_AddressBase [29/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S208C8.h.

5.1.2.2849 PORTC_AddressBase [30/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6269.h.

5.1.2.2850 PORTC_AddressBase [31/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6223.h.

5.1.2.2851 PORTC_AddressBase [32/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6388.h.

5.1.2.2852 PORTC_AddressBase [33/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF52AA.h.

5.1.2.2853 PORTC_AddressBase [34/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207S8.h.

5.1.2.2854 PORTC_AddressBase [35/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6226.h.

5.1.2.2855 PORTC_AddressBase [36/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207C8.h.

5.1.2.2856 PORTC_AddressBase [37/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6286.h.

5.1.2.2857 PORTC_AddressBase [38/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S208R6.h.

5.1.2.2858 PORTC_AddressBase [39/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207CB.h.

5.1.2.2859 PORTC_AddressBase [40/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207M8.h.

5.1.2.2860 PORTC_AddressBase [41/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207K8.h.

5.1.2.2861 PORTC_AddressBase [42/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207R6.h.

5.1.2.2862 PORTC_AddressBase [43/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF52A6.h.

5.1.2.2863 PORTC_AddressBase [44/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S003F3.h.

5.1.2.2864 PORTC_AddressBase [45/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S003K3.h.

5.1.2.2865 PORTC_AddressBase [46/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S005C6.h.

5.1.2.2866 PORTC_AddressBase [47/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S005K6.h.

5.1.2.2867 PORTC_AddressBase [48/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6289.h.

5.1.2.2868 PORTC_AddressBase [49/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S007C8.h.

5.1.2.2869 PORTC_AddressBase [50/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S105C4.h.

5.1.2.2870 PORTC_AddressBase [51/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF628A.h.

5.1.2.2871 PORTC_AddressBase [52/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S105S4.h.

5.1.2.2872 PORTC_AddressBase [53/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF5286.h.

5.1.2.2873 PORTC_AddressBase [54/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S208C6.h.

5.1.2.2874 PORTC_AddressBase [55/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S208RB.h.

5.1.2.2875 PORTC_AddressBase [56/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S105S6.h.

5.1.2.2876 PORTC_AddressBase [57/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF62A9.h.

5.1.2.2877 PORTC_AddressBase [58/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S105C6.h.

5.1.2.2878 PORTC_AddressBase [59/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207R8.h.

5.1.2.2879 PORTC_AddressBase [60/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF62A6.h.

5.1.2.2880 PORTC_AddressBase [61/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207S6.h.

5.1.2.2881 PORTC_AddressBase [62/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S105K4.h.

5.1.2.2882 PORTC_AddressBase [63/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207SB.h.

5.1.2.2883 PORTC_AddressBase [64/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6213A.h.

5.1.2.2884 PORTC_AddressBase [65/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S105K6.h.

5.1.2.2885 PORTC_AddressBase [66/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF6366.h.

5.1.2.2886 PORTC_AddressBase [67/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S208CB.h.

5.1.2.2887 PORTC_AddressBase [68/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S207C6.h.

5.1.2.2888 PORTC_AddressBase [69/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8S103F3.h.

5.1.2.2889 PORTC_AddressBase [70/70]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8AF528A.h.

5.1.2.2890 PORTD_AddressBase [1/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S103K3.h.

5.1.2.2891 PORTD_AddressBase [2/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF52A8.h.

5.1.2.2892 PORTD_AddressBase [3/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S903K3.h.

5.1.2.2893 PORTD_AddressBase [4/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S005K6.h.

5.1.2.2894 PORTD_AddressBase [5/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF5269.h.

5.1.2.2895 PORTD_AddressBase [6/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S007C8.h.

5.1.2.2896 PORTD_AddressBase [7/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S005C6.h.

5.1.2.2897 PORTD_AddressBase [8/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6289.h.

5.1.2.2898 PORTD_AddressBase [9/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6213.h.

5.1.2.2899 PORTD_AddressBase [10/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207RB.h.

5.1.2.2900 PORTD_AddressBase [11/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S003K3.h.

5.1.2.2901 PORTD_AddressBase [12/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6248.h.

5.1.2.2902 PORTD_AddressBase [13/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF5268.h.

5.1.2.2903 PORTD_AddressBase [14/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S001J3.h.

5.1.2.2904 PORTD_AddressBase [15/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF62A9.h.

5.1.2.2905 PORTD_AddressBase [16/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207S6.h.

5.1.2.2906 PORTD_AddressBase [17/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S003F3.h.

5.1.2.2907 PORTD_AddressBase [18/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6223A.h.

5.1.2.2908 PORTD_AddressBase [19/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6266.h.

5.1.2.2909 PORTD_AddressBase [20/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6226.h.

5.1.2.2910 PORTD_AddressBase [21/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF62A8.h.

5.1.2.2911 PORTD_AddressBase [22/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF52A9.h.

5.1.2.2912 PORTD_AddressBase [23/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S903F3.h.

5.1.2.2913 PORTD_AddressBase [24/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6246.h.

5.1.2.2914 PORTD_AddressBase [25/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S208SB.h.

5.1.2.2915 PORTD_AddressBase [26/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207SB.h.

5.1.2.2916 PORTD_AddressBase [27/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF5289.h.

5.1.2.2917 PORTD_AddressBase [28/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6268.h.

5.1.2.2918 PORTD_AddressBase [29/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S208C6.h.

5.1.2.2919 PORTD_AddressBase [30/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S208CB.h.

5.1.2.2920 PORTD_AddressBase [31/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S208R8.h.

5.1.2.2921 PORTD_AddressBase [32/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF5288.h.

5.1.2.2922 PORTD_AddressBase [33/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6366.h.

5.1.2.2923 PORTD_AddressBase [34/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF52AA.h.

5.1.2.2924 PORTD_AddressBase [35/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S208R6.h.

5.1.2.2925 PORTD_AddressBase [36/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6286.h.

5.1.2.2926 PORTD_AddressBase [37/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207CB.h.

5.1.2.2927 PORTD_AddressBase [38/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207C8.h.

5.1.2.2928 PORTD_AddressBase [39/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207K6.h.

5.1.2.2929 PORTD_AddressBase [40/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6223.h.

5.1.2.2930 PORTD_AddressBase [41/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207M8.h.

5.1.2.2931 PORTD_AddressBase [42/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207R6.h.

5.1.2.2932 PORTD_AddressBase [43/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207K8.h.

5.1.2.2933 PORTD_AddressBase [44/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6288.h.

5.1.2.2934 PORTD_AddressBase [45/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207MB.h.

5.1.2.2935 PORTD_AddressBase [46/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207S8.h.

5.1.2.2936 PORTD_AddressBase [47/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S208RB.h.

5.1.2.2937 PORTD_AddressBase [48/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207R8.h.

5.1.2.2938 PORTD_AddressBase [49/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S103F2.h.

5.1.2.2939 PORTD_AddressBase [50/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF5286.h.

5.1.2.2940 PORTD_AddressBase [51/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF628A.h.

5.1.2.2941 PORTD_AddressBase [52/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S208MB.h.

5.1.2.2942 PORTD_AddressBase [53/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6269.h.

5.1.2.2943 PORTD_AddressBase [54/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6388.h.

5.1.2.2944 PORTD_AddressBase [55/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S105K6.h.

5.1.2.2945 PORTD_AddressBase [56/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S208S6.h.

5.1.2.2946 PORTD_AddressBase [57/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S208S8.h.

5.1.2.2947 PORTD_AddressBase [58/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF528A.h.

5.1.2.2948 PORTD_AddressBase [59/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S105C4.h.

5.1.2.2949 PORTD_AddressBase [60/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF62AA.h.

5.1.2.2950 PORTD_AddressBase [61/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S105C6.h.

5.1.2.2951 PORTD_AddressBase [62/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF52A6.h.

5.1.2.2952 PORTD_AddressBase [63/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF62A6.h.

5.1.2.2953 PORTD_AddressBase [64/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S105S4.h.

5.1.2.2954 PORTD_AddressBase [65/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S105S6.h.

5.1.2.2955 PORTD_AddressBase [66/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S105K4.h.

5.1.2.2956 PORTD_AddressBase [67/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S207C6.h.

5.1.2.2957 PORTD_AddressBase [68/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S208C8.h.

5.1.2.2958 PORTD_AddressBase [69/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8AF6213A.h.

5.1.2.2959 PORTD_AddressBase [70/70]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8S103F3.h.

5.1.2.2960 PORTE_AddressBase [1/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF528A.h.

5.1.2.2961 PORTE_AddressBase [2/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S005K6.h.

5.1.2.2962 PORTE_AddressBase [3/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF52A8.h.

5.1.2.2963 PORTE_AddressBase [4/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S903K3.h.

5.1.2.2964 PORTE_AddressBase [5/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF5269.h.

5.1.2.2965 PORTE_AddressBase [6/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6213.h.

5.1.2.2966 PORTE_AddressBase [7/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S005C6.h.

5.1.2.2967 PORTE_AddressBase [8/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S007C8.h.

5.1.2.2968 PORTE_AddressBase [9/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6248.h.

5.1.2.2969 PORTE_AddressBase [10/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6288.h.

5.1.2.2970 PORTE_AddressBase [11/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF62AA.h.

5.1.2.2971 PORTE_AddressBase [12/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S001J3.h.

5.1.2.2972 PORTE_AddressBase [13/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6266.h.

5.1.2.2973 PORTE_AddressBase [14/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF52A9.h.

5.1.2.2974 PORTE_AddressBase [15/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S208C8.h.

5.1.2.2975 PORTE_AddressBase [16/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S903F3.h.

5.1.2.2976 PORTE_AddressBase [17/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S208S8.h.

5.1.2.2977 PORTE_AddressBase [18/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6268.h.

5.1.2.2978 PORTE_AddressBase [19/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S208SB.h.

5.1.2.2979 PORTE_AddressBase [20/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF5289.h.

5.1.2.2980 PORTE_AddressBase [21/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S208CB.h.

5.1.2.2981 PORTE_AddressBase [22/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF5288.h.

5.1.2.2982 PORTE_AddressBase [23/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6246.h.

5.1.2.2983 PORTE_AddressBase [24/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6269.h.

5.1.2.2984 PORTE_AddressBase [25/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S208RB.h.

5.1.2.2985 PORTE_AddressBase [26/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S208MB.h.

5.1.2.2986 PORTE_AddressBase [27/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6366.h.

5.1.2.2987 PORTE_AddressBase [28/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF5286.h.

5.1.2.2988 PORTE_AddressBase [29/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF62A8.h.

5.1.2.2989 PORTE_AddressBase [30/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF52AA.h.

5.1.2.2990 PORTE_AddressBase [31/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6286.h.

5.1.2.2991 PORTE_AddressBase [32/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207CB.h.

5.1.2.2992 PORTE_AddressBase [33/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6226.h.

5.1.2.2993 PORTE_AddressBase [34/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6388.h.

5.1.2.2994 PORTE_AddressBase [35/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207K6.h.

5.1.2.2995 PORTE_AddressBase [36/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF5268.h.

5.1.2.2996 PORTE_AddressBase [37/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S003F3.h.

5.1.2.2997 PORTE_AddressBase [38/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF62A9.h.

5.1.2.2998 PORTE_AddressBase [39/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S003K3.h.

5.1.2.2999 PORTE_AddressBase [40/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S208R6.h.

5.1.2.3000 PORTE_AddressBase [41/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6289.h.

5.1.2.3001 PORTE_AddressBase [42/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S208S6.h.

5.1.2.3002 PORTE_AddressBase [43/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S103F3.h.

5.1.2.3003 PORTE_AddressBase [44/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207S6.h.

5.1.2.3004 PORTE_AddressBase [45/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S105K4.h.

5.1.2.3005 PORTE_AddressBase [46/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207C6.h.

5.1.2.3006 PORTE_AddressBase [47/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207C8.h.

5.1.2.3007 PORTE_AddressBase [48/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S208R8.h.

5.1.2.3008 PORTE_AddressBase [49/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6223.h.

5.1.2.3009 PORTE_AddressBase [50/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207K8.h.

5.1.2.3010 PORTE_AddressBase [51/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207M8.h.

5.1.2.3011 PORTE_AddressBase [52/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S105S6.h.

5.1.2.3012 PORTE_AddressBase [53/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207MB.h.

5.1.2.3013 PORTE_AddressBase [54/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207R6.h.

5.1.2.3014 PORTE_AddressBase [55/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF62A6.h.

5.1.2.3015 PORTE_AddressBase [56/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S105C4.h.

5.1.2.3016 PORTE_AddressBase [57/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207R8.h.

5.1.2.3017 PORTE_AddressBase [58/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF52A6.h.

5.1.2.3018 PORTE_AddressBase [59/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S105S4.h.

5.1.2.3019 PORTE_AddressBase [60/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207RB.h.

5.1.2.3020 PORTE_AddressBase [61/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6223A.h.

5.1.2.3021 PORTE_AddressBase [62/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF628A.h.

5.1.2.3022 PORTE_AddressBase [63/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207S8.h.

5.1.2.3023 PORTE_AddressBase [64/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S207SB.h.

5.1.2.3024 PORTE_AddressBase [65/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8AF6213A.h.

5.1.2.3025 PORTE_AddressBase [66/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S208C6.h.

5.1.2.3026 PORTE_AddressBase [67/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S105K6.h.

5.1.2.3027 PORTE_AddressBase [68/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S103K3.h.

5.1.2.3028 PORTE_AddressBase [69/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S105C6.h.

5.1.2.3029 PORTE_AddressBase [70/70]

```
#define PORTE_AddressBase 0x5014
```

Definition at line 57 of file STM8S103F2.h.

5.1.2.3030 PORTF_AddressBase [1/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6266.h.

5.1.2.3031 PORTF_AddressBase [2/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6213.h.

5.1.2.3032 PORTF_AddressBase [3/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S903K3.h.

5.1.2.3033 PORTF_AddressBase [4/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF52A8.h.

5.1.2.3034 PORTF_AddressBase [5/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S007C8.h.

5.1.2.3035 PORTF_AddressBase [6/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S005K6.h.

5.1.2.3036 PORTF_AddressBase [7/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207RB.h.

5.1.2.3037 PORTF_AddressBase [8/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S005C6.h.

5.1.2.3038 PORTF_AddressBase [9/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6289.h.

5.1.2.3039 PORTF_AddressBase [10/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207R8.h.

5.1.2.3040 PORTF_AddressBase [11/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6248.h.

5.1.2.3041 PORTF_AddressBase [12/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S001J3.h.

5.1.2.3042 PORTF_AddressBase [13/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6223A.h.

5.1.2.3043 PORTF_AddressBase [14/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF62AA.h.

5.1.2.3044 PORTF_AddressBase [15/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S003F3.h.

5.1.2.3045 PORTF_AddressBase [16/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6288.h.

5.1.2.3046 PORTF_AddressBase [17/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207S6.h.

5.1.2.3047 PORTF_AddressBase [18/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF5268.h.

5.1.2.3048 PORTF_AddressBase [19/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207K6.h.

5.1.2.3049 PORTF_AddressBase [20/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207MB.h.

5.1.2.3050 PORTF_AddressBase [21/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S903F3.h.

5.1.2.3051 PORTF_AddressBase [22/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF52A9.h.

5.1.2.3052 PORTF_AddressBase [23/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S208C6.h.

5.1.2.3053 PORTF_AddressBase [24/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207SB.h.

5.1.2.3054 PORTF_AddressBase [25/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S208SB.h.

5.1.2.3055 PORTF_AddressBase [26/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S103K3.h.

5.1.2.3056 PORTF_AddressBase [27/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S208R8.h.

5.1.2.3057 PORTF_AddressBase [28/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S208S8.h.

5.1.2.3058 PORTF_AddressBase [29/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6269.h.

5.1.2.3059 PORTF_AddressBase [30/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S208C8.h.

5.1.2.3060 PORTF_AddressBase [31/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S208RB.h.

5.1.2.3061 PORTF_AddressBase [32/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF5288.h.

5.1.2.3062 PORTF_AddressBase [33/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6366.h.

5.1.2.3063 PORTF_AddressBase [34/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S208CB.h.

5.1.2.3064 PORTF_AddressBase [35/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF52AA.h.

5.1.2.3065 PORTF_AddressBase [36/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF5286.h.

5.1.2.3066 PORTF_AddressBase [37/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S208R6.h.

5.1.2.3067 PORTF_AddressBase [38/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207C8.h.

5.1.2.3068 PORTF_AddressBase [39/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207CB.h.

5.1.2.3069 PORTF_AddressBase [40/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207R6.h.

5.1.2.3070 PORTF_AddressBase [41/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF62A9.h.

5.1.2.3071 PORTF_AddressBase [42/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207M8.h.

5.1.2.3072 PORTF_AddressBase [43/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6286.h.

5.1.2.3073 PORTF_AddressBase [44/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207K8.h.

5.1.2.3074 PORTF_AddressBase [45/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S003K3.h.

5.1.2.3075 PORTF_AddressBase [46/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF528A.h.

5.1.2.3076 PORTF_AddressBase [47/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF62A6.h.

5.1.2.3077 PORTF_AddressBase [48/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S105S6.h.

5.1.2.3078 PORTF_AddressBase [49/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF5289.h.

5.1.2.3079 PORTF_AddressBase [50/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207S8.h.

5.1.2.3080 PORTF_AddressBase [51/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF62A8.h.

5.1.2.3081 PORTF_AddressBase [52/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6223.h.

5.1.2.3082 PORTF_AddressBase [53/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6268.h.

5.1.2.3083 PORTF_AddressBase [54/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6246.h.

5.1.2.3084 PORTF_AddressBase [55/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S105C4.h.

5.1.2.3085 PORTF_AddressBase [56/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S105S4.h.

5.1.2.3086 PORTF_AddressBase [57/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF628A.h.

5.1.2.3087 PORTF_AddressBase [58/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF52A6.h.

5.1.2.3088 PORTF_AddressBase [59/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S105K4.h.

5.1.2.3089 PORTF_AddressBase [60/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S105K6.h.

5.1.2.3090 PORTF_AddressBase [61/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6213A.h.

5.1.2.3091 PORTF_AddressBase [62/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S105C6.h.

5.1.2.3092 PORTF_AddressBase [63/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S207C6.h.

5.1.2.3093 PORTF_AddressBase [64/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S208MB.h.

5.1.2.3094 PORTF_AddressBase [65/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6226.h.

5.1.2.3095 PORTF_AddressBase [66/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF6388.h.

5.1.2.3096 PORTF_AddressBase [67/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S103F3.h.

5.1.2.3097 PORTF_AddressBase [68/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8AF5269.h.

5.1.2.3098 PORTF_AddressBase [69/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S103F2.h.

5.1.2.3099 PORTF_AddressBase [70/70]

```
#define PORTF_AddressBase 0x5019
```

Definition at line 58 of file STM8S208S6.h.

5.1.2.3100 PORTG_AddressBase [1/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207CB.h.

5.1.2.3101 PORTG_AddressBase [2/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S005K6.h.

5.1.2.3102 PORTG_AddressBase [3/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207RB.h.

5.1.2.3103 PORTG_AddressBase [4/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S005C6.h.

5.1.2.3104 PORTG_AddressBase [5/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207R8.h.

5.1.2.3105 PORTG_AddressBase [6/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF6248.h.

5.1.2.3106 PORTG_AddressBase [7/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF6289.h.

5.1.2.3107 PORTG_AddressBase [8/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207K6.h.

5.1.2.3108 PORTG_AddressBase [9/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF62AA.h.

5.1.2.3109 PORTG_AddressBase [10/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207S6.h.

5.1.2.3110 PORTG_AddressBase [11/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207S8.h.

5.1.2.3111 PORTG_AddressBase [12/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S208CB.h.

5.1.2.3112 PORTG_AddressBase [13/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF528A.h.

5.1.2.3113 PORTG_AddressBase [14/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF6266.h.

5.1.2.3114 PORTG_AddressBase [15/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF5286.h.

5.1.2.3115 PORTG_AddressBase [16/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF6268.h.

5.1.2.3116 PORTG_AddressBase [17/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF5289.h.

5.1.2.3117 PORTG_AddressBase [18/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S208R8.h.

5.1.2.3118 PORTG_AddressBase [19/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF6246.h.

5.1.2.3119 PORTG_AddressBase [20/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF52A9.h.

5.1.2.3120 PORTG_AddressBase [21/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF6269.h.

5.1.2.3121 PORTG_AddressBase [22/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF5288.h.

5.1.2.3122 PORTG_AddressBase [23/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207C8.h.

5.1.2.3123 PORTG_AddressBase [24/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF52AA.h.

5.1.2.3124 PORTG_AddressBase [25/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S208R6.h.

5.1.2.3125 PORTG_AddressBase [26/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF62A8.h.

5.1.2.3126 PORTG_AddressBase [27/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF6286.h.

5.1.2.3127 PORTG_AddressBase [28/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207MB.h.

5.1.2.3128 PORTG_AddressBase [29/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207R6.h.

5.1.2.3129 PORTG_AddressBase [30/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207K8.h.

5.1.2.3130 PORTG_AddressBase [31/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF6288.h.

5.1.2.3131 PORTG_AddressBase [32/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S007C8.h.

5.1.2.3132 PORTG_AddressBase [33/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF6388.h.

5.1.2.3133 PORTG_AddressBase [34/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S208MB.h.

5.1.2.3134 PORTG_AddressBase [35/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S208S6.h.

5.1.2.3135 PORTG_AddressBase [36/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S208C8.h.

5.1.2.3136 PORTG_AddressBase [37/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF62A9.h.

5.1.2.3137 PORTG_AddressBase [38/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF5268.h.

5.1.2.3138 PORTG_AddressBase [39/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S208SB.h.

5.1.2.3139 PORTG_AddressBase [40/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207SB.h.

5.1.2.3140 PORTG_AddressBase [41/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF628A.h.

5.1.2.3141 PORTG_AddressBase [42/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207M8.h.

5.1.2.3142 PORTG_AddressBase [43/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF62A6.h.

5.1.2.3143 PORTG_AddressBase [44/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S105S6.h.

5.1.2.3144 PORTG_AddressBase [45/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF5269.h.

5.1.2.3145 PORTG_AddressBase [46/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S105C4.h.

5.1.2.3146 PORTG_AddressBase [47/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF52A6.h.

5.1.2.3147 PORTG_AddressBase [48/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S105S4.h.

5.1.2.3148 PORTG_AddressBase [49/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S207C6.h.

5.1.2.3149 PORTG_AddressBase [50/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S208C6.h.

5.1.2.3150 PORTG_AddressBase [51/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S105K6.h.

5.1.2.3151 PORTG_AddressBase [52/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8AF52A8.h.

5.1.2.3152 PORTG_AddressBase [53/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S105K4.h.

5.1.2.3153 PORTG_AddressBase [54/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S105C6.h.

5.1.2.3154 PORTG_AddressBase [55/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S208RB.h.

5.1.2.3155 PORTG_AddressBase [56/56]

```
#define PORTG_AddressBase 0x501E
```

Definition at line 59 of file STM8S208S8.h.

5.1.2.3156 PORTH_AddressBase [1/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S208SB.h.

5.1.2.3157 PORTH_AddressBase [2/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF52A8.h.

5.1.2.3158 PORTH_AddressBase [3/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF62A9.h.

5.1.2.3159 PORTH_AddressBase [4/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207R8.h.

5.1.2.3160 PORTH_AddressBase [5/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207RB.h.

5.1.2.3161 PORTH_AddressBase [6/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF52A6.h.

5.1.2.3162 PORTH_AddressBase [7/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207S6.h.

5.1.2.3163 PORTH_AddressBase [8/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207S8.h.

5.1.2.3164 PORTH_AddressBase [9/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207K6.h.

5.1.2.3165 PORTH_AddressBase [10/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF5286.h.

5.1.2.3166 PORTH_AddressBase [11/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF52A9.h.

5.1.2.3167 PORTH_AddressBase [12/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF5268.h.

5.1.2.3168 PORTH_AddressBase [13/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF5289.h.

5.1.2.3169 PORTH_AddressBase [14/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF6388.h.

5.1.2.3170 PORTH_AddressBase [15/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207SB.h.

5.1.2.3171 PORTH_AddressBase [16/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S208S6.h.

5.1.2.3172 PORTH_AddressBase [17/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S208MB.h.

5.1.2.3173 PORTH_AddressBase [18/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207C8.h.

5.1.2.3174 PORTH_AddressBase [19/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207MB.h.

5.1.2.3175 PORTH_AddressBase [20/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207M8.h.

5.1.2.3176 PORTH_AddressBase [21/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207R6.h.

5.1.2.3177 PORTH_AddressBase [22/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF628A.h.

5.1.2.3178 PORTH_AddressBase [23/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF6288.h.

5.1.2.3179 PORTH_AddressBase [24/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S208C6.h.

5.1.2.3180 PORTH_AddressBase [25/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S208R6.h.

5.1.2.3181 PORTH_AddressBase [26/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF5288.h.

5.1.2.3182 PORTH_AddressBase [27/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF52AA.h.

5.1.2.3183 PORTH_AddressBase [28/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF6286.h.

5.1.2.3184 PORTH_AddressBase [29/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF62A8.h.

5.1.2.3185 PORTH_AddressBase [30/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S208RB.h.

5.1.2.3186 PORTH_AddressBase [31/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S208CB.h.

5.1.2.3187 PORTH_AddressBase [32/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S208S8.h.

5.1.2.3188 PORTH_AddressBase [33/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF6289.h.

5.1.2.3189 PORTH_AddressBase [34/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207CB.h.

5.1.2.3190 PORTH_AddressBase [35/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF62AA.h.

5.1.2.3191 PORTH_AddressBase [36/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207C6.h.

5.1.2.3192 PORTH_AddressBase [37/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF5269.h.

5.1.2.3193 PORTH_AddressBase [38/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S208C8.h.

5.1.2.3194 PORTH_AddressBase [39/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF62A6.h.

5.1.2.3195 PORTH_AddressBase [40/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S207K8.h.

5.1.2.3196 PORTH_AddressBase [41/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S007C8.h.

5.1.2.3197 PORTH_AddressBase [42/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8S208R8.h.

5.1.2.3198 PORTH_AddressBase [43/43]

```
#define PORTH_AddressBase 0x5023
```

Definition at line 60 of file STM8AF528A.h.

5.1.2.3199 PORTI_AddressBase [1/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF6388.h.

5.1.2.3200 PORTI_AddressBase [2/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF5288.h.

5.1.2.3201 PORTI_AddressBase [3/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF6288.h.

5.1.2.3202 PORTI_AddressBase [4/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207R8.h.

5.1.2.3203 PORTI_AddressBase [5/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207S6.h.

5.1.2.3204 PORTI_AddressBase [6/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207K6.h.

5.1.2.3205 PORTI_AddressBase [7/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF62AA.h.

5.1.2.3206 PORTI_AddressBase [8/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S208R6.h.

5.1.2.3207 PORTI_AddressBase [9/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207CB.h.

5.1.2.3208 PORTI_AddressBase [10/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF52A6.h.

5.1.2.3209 PORTI_AddressBase [11/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207S8.h.

5.1.2.3210 PORTI_AddressBase [12/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207MB.h.

5.1.2.3211 PORTI_AddressBase [13/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207M8.h.

5.1.2.3212 PORTI_AddressBase [14/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207SB.h.

5.1.2.3213 PORTI_AddressBase [15/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF5289.h.

5.1.2.3214 PORTI_AddressBase [16/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S208R8.h.

5.1.2.3215 PORTI_AddressBase [17/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF52A9.h.

5.1.2.3216 PORTI_AddressBase [18/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207R6.h.

5.1.2.3217 PORTI_AddressBase [19/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S208C8.h.

5.1.2.3218 PORTI_AddressBase [20/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S208MB.h.

5.1.2.3219 PORTI_AddressBase [21/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF6286.h.

5.1.2.3220 PORTI_AddressBase [22/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF62A9.h.

5.1.2.3221 PORTI_AddressBase [23/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF52AA.h.

5.1.2.3222 PORTI_AddressBase [24/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S208S6.h.

5.1.2.3223 PORTI_AddressBase [25/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207C8.h.

5.1.2.3224 PORTI_AddressBase [26/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF528A.h.

5.1.2.3225 PORTI_AddressBase [27/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF628A.h.

5.1.2.3226 PORTI_AddressBase [28/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF52A8.h.

5.1.2.3227 PORTI_AddressBase [29/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207RB.h.

5.1.2.3228 PORTI_AddressBase [30/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S208RB.h.

5.1.2.3229 PORTI_AddressBase [31/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF62A6.h.

5.1.2.3230 PORTI_AddressBase [32/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S208CB.h.

5.1.2.3231 PORTI_AddressBase [33/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S208SB.h.

5.1.2.3232 PORTI_AddressBase [34/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF5269.h.

5.1.2.3233 PORTI_AddressBase [35/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF62A8.h.

5.1.2.3234 PORTI_AddressBase [36/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S208S8.h.

5.1.2.3235 PORTI_AddressBase [37/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207K8.h.

5.1.2.3236 PORTI_AddressBase [38/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S208C6.h.

5.1.2.3237 PORTI_AddressBase [39/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S007C8.h.

5.1.2.3238 PORTI_AddressBase [40/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF5286.h.

5.1.2.3239 PORTI_AddressBase [41/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF6289.h.

5.1.2.3240 PORTI_AddressBase [42/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8AF5268.h.

5.1.2.3241 PORTI_AddressBase [43/43]

```
#define PORTI_AddressBase 0x5028
```

Definition at line 61 of file STM8S207C6.h.

5.1.2.3242 RST_AddressBase [1/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8S103K3.h.

5.1.2.3243 RST_AddressBase [2/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8AF6213.h.

5.1.2.3244 RST_AddressBase [3/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8S001J3.h.

5.1.2.3245 RST_AddressBase [4/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8S003F3.h.

5.1.2.3246 RST_AddressBase [5/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8AF6223A.h.

5.1.2.3247 RST_AddressBase [6/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8AF6366.h.

5.1.2.3248 RST_AddressBase [7/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8S903K3.h.

5.1.2.3249 RST_AddressBase [8/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8AF6226.h.

5.1.2.3250 RST_AddressBase [9/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8AF6223.h.

5.1.2.3251 RST_AddressBase [10/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8S003K3.h.

5.1.2.3252 RST_AddressBase [11/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8S903F3.h.

5.1.2.3253 RST_AddressBase [12/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8S103F2.h.

5.1.2.3254 RST_AddressBase [13/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8AF6213A.h.

5.1.2.3255 RST_AddressBase [14/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 61 of file STM8S103F3.h.

5.1.2.3256 RST_AddressBase [15/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8AF6246.h.

5.1.2.3257 RST_AddressBase [16/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8AF6248.h.

5.1.2.3258 RST_AddressBase [17/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8AF6268.h.

5.1.2.3259 RST_AddressBase [18/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8S005C6.h.

5.1.2.3260 RST_AddressBase [19/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8AF6269.h.

5.1.2.3261 RST_AddressBase [20/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8S105S4.h.

5.1.2.3262 RST_AddressBase [21/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8S105S6.h.

5.1.2.3263 RST_AddressBase [22/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8AF6266.h.

5.1.2.3264 RST_AddressBase [23/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8S105K4.h.

5.1.2.3265 RST_AddressBase [24/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8S105C4.h.

5.1.2.3266 RST_AddressBase [25/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8S105C6.h.

5.1.2.3267 RST_AddressBase [26/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8S005K6.h.

5.1.2.3268 RST_AddressBase [27/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 62 of file STM8S105K6.h.

5.1.2.3269 RST_AddressBase [28/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF62A8.h.

5.1.2.3270 RST_AddressBase [29/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207R8.h.

5.1.2.3271 RST_AddressBase [30/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207MB.h.

5.1.2.3272 RST_AddressBase [31/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207S6.h.

5.1.2.3273 RST_AddressBase [32/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF6288.h.

5.1.2.3274 RST_AddressBase [33/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207S8.h.

5.1.2.3275 RST_AddressBase [34/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF62AA.h.

5.1.2.3276 RST_AddressBase [35/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF52AA.h.

5.1.2.3277 RST_AddressBase [36/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF5269.h.

5.1.2.3278 RST_AddressBase [37/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S208SB.h.

5.1.2.3279 RST_AddressBase [38/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF52A9.h.

5.1.2.3280 RST_AddressBase [39/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207SB.h.

5.1.2.3281 RST_AddressBase [40/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207R6.h.

5.1.2.3282 RST_AddressBase [41/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S208S8.h.

5.1.2.3283 RST_AddressBase [42/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF5268.h.

5.1.2.3284 RST_AddressBase [43/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF6388.h.

5.1.2.3285 RST_AddressBase [44/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF52A8.h.

5.1.2.3286 RST_AddressBase [45/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S208C8.h.

5.1.2.3287 RST_AddressBase [46/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S208R6.h.

5.1.2.3288 RST_AddressBase [47/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S208CB.h.

5.1.2.3289 RST_AddressBase [48/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S208RB.h.

5.1.2.3290 RST_AddressBase [49/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF5288.h.

5.1.2.3291 RST_AddressBase [50/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S208MB.h.

5.1.2.3292 RST_AddressBase [51/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF6286.h.

5.1.2.3293 RST_AddressBase [52/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207M8.h.

5.1.2.3294 RST_AddressBase [53/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207K6.h.

5.1.2.3295 RST_AddressBase [54/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207K8.h.

5.1.2.3296 RST_AddressBase [55/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF62A9.h.

5.1.2.3297 RST_AddressBase [56/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF5289.h.

5.1.2.3298 RST_AddressBase [57/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207C6.h.

5.1.2.3299 RST_AddressBase [58/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF5286.h.

5.1.2.3300 RST_AddressBase [59/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF6289.h.

5.1.2.3301 RST_AddressBase [60/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S208S6.h.

5.1.2.3302 RST_AddressBase [61/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207C8.h.

5.1.2.3303 RST_AddressBase [62/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207RB.h.

5.1.2.3304 RST_AddressBase [63/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S208R8.h.

5.1.2.3305 RST_AddressBase [64/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S208C6.h.

5.1.2.3306 RST_AddressBase [65/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF528A.h.

5.1.2.3307 RST_AddressBase [66/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S207CB.h.

5.1.2.3308 RST_AddressBase [67/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF628A.h.

5.1.2.3309 RST_AddressBase [68/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF62A6.h.

5.1.2.3310 RST_AddressBase [69/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8AF52A6.h.

5.1.2.3311 RST_AddressBase [70/70]

```
#define RST_AddressBase 0x50B3
```

Definition at line 64 of file STM8S007C8.h.

5.1.2.3312 SPI_AddressBase [1/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8S003F3.h.

5.1.2.3313 SPI_AddressBase [2/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8S903K3.h.

5.1.2.3314 SPI_AddressBase [3/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8AF6223A.h.

5.1.2.3315 SPI_AddressBase [4/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8S001J3.h.

5.1.2.3316 SPI_AddressBase [5/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8S003K3.h.

5.1.2.3317 SPI_AddressBase [6/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8S103K3.h.

5.1.2.3318 SPI_AddressBase [7/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8S103F2.h.

5.1.2.3319 SPI_AddressBase [8/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8AF6223.h.

5.1.2.3320 SPI_AddressBase [9/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8AF6213A.h.

5.1.2.3321 SPI_AddressBase [10/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8AF6366.h.

5.1.2.3322 SPI_AddressBase [11/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8S903F3.h.

5.1.2.3323 SPI_AddressBase [12/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8AF6226.h.

5.1.2.3324 SPI_AddressBase [13/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8S103F3.h.

5.1.2.3325 SPI_AddressBase [14/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 67 of file STM8AF6213.h.

5.1.2.3326 SPI_AddressBase [15/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8S005C6.h.

5.1.2.3327 SPI_AddressBase [16/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8AF6248.h.

5.1.2.3328 SPI_AddressBase [17/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8AF6266.h.

5.1.2.3329 SPI_AddressBase [18/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8S005K6.h.

5.1.2.3330 SPI_AddressBase [19/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8AF6268.h.

5.1.2.3331 SPI_AddressBase [20/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8AF6246.h.

5.1.2.3332 SPI_AddressBase [21/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8AF6269.h.

5.1.2.3333 SPI_AddressBase [22/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8S105S6.h.

5.1.2.3334 SPI_AddressBase [23/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8S105K4.h.

5.1.2.3335 SPI_AddressBase [24/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8S105K6.h.

5.1.2.3336 SPI_AddressBase [25/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8S105S4.h.

5.1.2.3337 SPI_AddressBase [26/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8S105C4.h.

5.1.2.3338 SPI_AddressBase [27/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 68 of file STM8S105C6.h.

5.1.2.3339 SPI_AddressBase [28/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF6289.h.

5.1.2.3340 SPI_AddressBase [29/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S208MB.h.

5.1.2.3341 SPI_AddressBase [30/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF5268.h.

5.1.2.3342 SPI_AddressBase [31/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207RB.h.

5.1.2.3343 SPI_AddressBase [32/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF52A8.h.

5.1.2.3344 SPI_AddressBase [33/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF6288.h.

5.1.2.3345 SPI_AddressBase [34/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207R8.h.

5.1.2.3346 SPI_AddressBase [35/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207S6.h.

5.1.2.3347 SPI_AddressBase [36/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207M8.h.

5.1.2.3348 SPI_AddressBase [37/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF6286.h.

5.1.2.3349 SPI_AddressBase [38/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF52AA.h.

5.1.2.3350 SPI_AddressBase [39/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S208R6.h.

5.1.2.3351 SPI_AddressBase [40/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207C6.h.

5.1.2.3352 SPI_AddressBase [41/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207SB.h.

5.1.2.3353 SPI_AddressBase [42/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF62AA.h.

5.1.2.3354 SPI_AddressBase [43/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S208C6.h.

5.1.2.3355 SPI_AddressBase [44/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207S8.h.

5.1.2.3356 SPI_AddressBase [45/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S208SB.h.

5.1.2.3357 SPI_AddressBase [46/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207C8.h.

5.1.2.3358 SPI_AddressBase [47/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S208C8.h.

5.1.2.3359 SPI_AddressBase [48/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207K8.h.

5.1.2.3360 SPI_AddressBase [49/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207MB.h.

5.1.2.3361 SPI_AddressBase [50/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207K6.h.

5.1.2.3362 SPI_AddressBase [51/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF5288.h.

5.1.2.3363 SPI_AddressBase [52/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF5269.h.

5.1.2.3364 SPI_AddressBase [53/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207R6.h.

5.1.2.3365 SPI_AddressBase [54/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF5289.h.

5.1.2.3366 SPI_AddressBase [55/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S208S6.h.

5.1.2.3367 SPI_AddressBase [56/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S208RB.h.

5.1.2.3368 SPI_AddressBase [57/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S207CB.h.

5.1.2.3369 SPI_AddressBase [58/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF62A6.h.

5.1.2.3370 SPI_AddressBase [59/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S208R8.h.

5.1.2.3371 SPI_AddressBase [60/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF52A9.h.

5.1.2.3372 SPI_AddressBase [61/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S208S8.h.

5.1.2.3373 SPI_AddressBase [62/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF62A8.h.

5.1.2.3374 SPI_AddressBase [63/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S208CB.h.

5.1.2.3375 SPI_AddressBase [64/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF528A.h.

5.1.2.3376 SPI_AddressBase [65/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF62A9.h.

5.1.2.3377 SPI_AddressBase [66/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF5286.h.

5.1.2.3378 SPI_AddressBase [67/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8S007C8.h.

5.1.2.3379 SPI_AddressBase [68/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF52A6.h.

5.1.2.3380 SPI_AddressBase [69/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF628A.h.

5.1.2.3381 SPI_AddressBase [70/70]

```
#define SPI_AddressBase 0x5200
```

Definition at line 70 of file STM8AF6388.h.

5.1.2.3382 STM8_ADDR_WIDTH

```
#define STM8_ADDR_WIDTH 16
```

width of address space

Definition at line 81 of file STM8AF_STM8S.h.

5.1.2.3383 STM8_EEPROM_END

```
#define STM8_EEPROM_END (STM8_EEPROM_START + STM8_EEPROM_SIZE - 1)
```

last address in EEPROM

Definition at line 77 of file STM8AF_STM8S.h.

5.1.2.3384 STM8_EEPROM_SIZE [1/71]

```
#define STM8_EEPROM_SIZE 1536
```

Definition at line 49 of file STM8S207S8.h.

5.1.2.3385 STM8_EEPROM_SIZE [2/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF6289.h.

5.1.2.3386 STM8_EEPROM_SIZE [3/71]

```
#define STM8_EEPROM_SIZE 1536
```

Definition at line 49 of file STM8S208SB.h.

5.1.2.3387 STM8_EEPROM_SIZE [4/71]

```
#define STM8_EEPROM_SIZE 640
```

Definition at line 49 of file STM8S903F3.h.

5.1.2.3388 STM8_EEPROM_SIZE [5/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8S207MB.h.

5.1.2.3389 STM8_EEPROM_SIZE [6/71]

```
#define STM8_EEPROM_SIZE 1536
```

Definition at line 49 of file STM8S207R8.h.

5.1.2.3390 STM8_EEPROM_SIZE [7/71]

```
#define STM8_EEPROM_SIZE 128
```

Definition at line 49 of file STM8S007C8.h.

5.1.2.3391 STM8_EEPROM_SIZE [8/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8S207RB.h.

5.1.2.3392 STM8_EEPROM_SIZE [9/71]

```
#define STM8_EEPROM_SIZE 640
```

Definition at line 49 of file STM8AF6213.h.

5.1.2.3393 STM8_EEPROM_SIZE [10/71]

```
#define STM8_EEPROM_SIZE 1536
```

Definition at line 49 of file STM8S208S6.h.

5.1.2.3394 STM8_EEPROM_SIZE [11/71]

```
#define STM8_EEPROM_SIZE 128
```

Definition at line 49 of file STM8S001J3.h.

5.1.2.3395 STM8_EEPROM_SIZE [12/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8S207K8.h.

5.1.2.3396 STM8_EEPROM_SIZE [13/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8AF5268.h.

5.1.2.3397 STM8_EEPROM_SIZE [14/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8S207R6.h.

5.1.2.3398 STM8_EEPROM_SIZE [15/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8AF6266.h.

5.1.2.3399 STM8_EEPROM_SIZE [16/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF6288.h.

5.1.2.3400 STM8_EEPROM_SIZE [17/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF5288.h.

5.1.2.3401 STM8_EEPROM_SIZE [18/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8S207S6.h.

5.1.2.3402 STM8_EEPROM_SIZE [19/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF5289.h.

5.1.2.3403 STM8_EEPROM_SIZE [20/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8AF6268.h.

5.1.2.3404 STM8_EEPROM_SIZE [21/71]

```
#define STM8_EEPROM_SIZE 512
```

Definition at line 49 of file STM8AF6246.h.

5.1.2.3405 STM8_EEPROM_SIZE [22/71]

```
#define STM8_EEPROM_SIZE 1536
```

Definition at line 49 of file STM8S207SB.h.

5.1.2.3406 STM8_EEPROM_SIZE [23/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF528A.h.

5.1.2.3407 STM8_EEPROM_SIZE [24/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8AF6366.h.

5.1.2.3408 STM8_EEPROM_SIZE [25/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF6388.h.

5.1.2.3409 STM8_EEPROM_SIZE [26/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8S208CB.h.

5.1.2.3410 STM8_EEPROM_SIZE [27/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8S208RB.h.

5.1.2.3411 STM8_EEPROM_SIZE [28/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF52AA.h.

5.1.2.3412 STM8_EEPROM_SIZE [29/71]

```
#define STM8_EEPROM_SIZE 1536
```

Definition at line 49 of file STM8S208S8.h.

5.1.2.3413 STM8_EEPROM_SIZE [30/71]

```
#define STM8_EEPROM_SIZE 128
```

Definition at line 49 of file STM8S005K6.h.

5.1.2.3414 STM8_EEPROM_SIZE [31/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8S207K6.h.

5.1.2.3415 STM8_EEPROM_SIZE [32/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8S207M8.h.

5.1.2.3416 STM8_EEPROM_SIZE [33/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8S207C6.h.

5.1.2.3417 STM8_EEPROM_SIZE [34/71]

```
#define STM8_EEPROM_SIZE 128
```

Definition at line 49 of file STM8S003K3.h.

5.1.2.3418 STM8_EEPROM_SIZE [35/71]

```
#define STM8_EEPROM_SIZE 128
```

Definition at line 49 of file STM8S005C6.h.

5.1.2.3419 STM8_EEPROM_SIZE [36/71]

```
#define STM8_EEPROM_SIZE 640
```

Definition at line 49 of file STM8S903K3.h.

5.1.2.3420 STM8_EEPROM_SIZE [37/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF628A.h.

5.1.2.3421 STM8_EEPROM_SIZE [38/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF62A8.h.

5.1.2.3422 STM8_EEPROM_SIZE [39/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF6286.h.

5.1.2.3423 STM8_EEPROM_SIZE [40/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8S208R6.h.

5.1.2.3424 STM8_EEPROM_SIZE [41/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8S208MB.h.

5.1.2.3425 STM8_EEPROM_SIZE [42/71]

```
#define STM8_EEPROM_SIZE 1536
```

Definition at line 49 of file STM8S207C8.h.

5.1.2.3426 STM8_EEPROM_SIZE [43/71]

```
#define STM8_EEPROM_SIZE 640
```

Definition at line 49 of file STM8S103F3.h.

5.1.2.3427 STM8_EEPROM_SIZE [44/71]

```
#define STM8_EEPROM_SIZE 640
```

Definition at line 49 of file STM8S103K3.h.

5.1.2.3428 STM8_EEPROM_SIZE [45/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8AF5269.h.

5.1.2.3429 STM8_EEPROM_SIZE [46/71]

```
#define STM8_EEPROM_SIZE 640
```

Definition at line 49 of file STM8AF6226.h.

5.1.2.3430 STM8_EEPROM_SIZE [47/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8S105C6.h.

5.1.2.3431 STM8_EEPROM_SIZE [48/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF52A6.h.

5.1.2.3432 STM8_EEPROM_SIZE [49/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8S207CB.h.

5.1.2.3433 STM8_EEPROM_SIZE [50/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8S105S6.h.

5.1.2.3434 STM8_EEPROM_SIZE [51/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8S208R8.h.

5.1.2.3435 STM8_EEPROM_SIZE [52/71]

```
#define STM8_EEPROM_SIZE 640
```

Definition at line 49 of file STM8AF6223A.h.

5.1.2.3436 STM8_EEPROM_SIZE [53/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8AF6269.h.

5.1.2.3437 STM8_EEPROM_SIZE [54/71]

```
#define STM8_EEPROM_SIZE 640
```

Definition at line 49 of file STM8AF6223.h.

5.1.2.3438 STM8_EEPROM_SIZE [55/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF5286.h.

5.1.2.3439 STM8_EEPROM_SIZE [56/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8S208C6.h.

5.1.2.3440 STM8_EEPROM_SIZE [57/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8S105K6.h.

5.1.2.3441 STM8_EEPROM_SIZE [58/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF52A8.h.

5.1.2.3442 STM8_EEPROM_SIZE [59/71]

```
#define STM8_EEPROM_SIZE 512
```

Definition at line 49 of file STM8AF6248.h.

5.1.2.3443 STM8_EEPROM_SIZE [60/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF62A6.h.

5.1.2.3444 STM8_EEPROM_SIZE [61/71]

```
#define STM8_EEPROM_SIZE 640
```

Definition at line 49 of file STM8AF6213A.h.

5.1.2.3445 STM8_EEPROM_SIZE [62/71]

```
#define STM8_EEPROM_SIZE 128
```

Definition at line 49 of file STM8S003F3.h.

5.1.2.3446 STM8_EEPROM_SIZE [63/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF52A9.h.

5.1.2.3447 STM8_EEPROM_SIZE [64/71]

```
#define STM8_EEPROM_SIZE 640
```

Definition at line 49 of file STM8S103F2.h.

5.1.2.3448 STM8_EEPROM_SIZE [65/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF62AA.h.

5.1.2.3449 STM8_EEPROM_SIZE [66/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8S105S4.h.

5.1.2.3450 STM8_EEPROM_SIZE [67/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8S105K4.h.

5.1.2.3451 STM8_EEPROM_SIZE [68/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8S208C8.h.

5.1.2.3452 STM8_EEPROM_SIZE [69/71]

```
#define STM8_EEPROM_SIZE 2048
```

Definition at line 49 of file STM8AF62A9.h.

5.1.2.3453 STM8_EEPROM_SIZE [70/71]

```
#define STM8_EEPROM_SIZE 1024
```

Definition at line 49 of file STM8S105C4.h.

5.1.2.3454 STM8_EEPROM_SIZE [71/71]

```
#define STM8_EEPROM_SIZE 128
```

size of data EEPROM [B]

Definition at line 68 of file STM8AF_STM8S.h.

5.1.2.3455 STM8_EEPROM_START

```
#define STM8_EEPROM_START 0x4000
```

first address in EEPROM

Definition at line 76 of file STM8AF_STM8S.h.

5.1.2.3456 STM8_MEM_POINTER_T

```
#define STM8_MEM_POINTER_T uint16_t
```

address variable type

Definition at line 82 of file STM8AF_STM8S.h.

5.1.2.3457 STM8_PFLASH_END

```
#define STM8_PFLASH_END (STM8_PFLASH_START + STM8_PFLASH_SIZE - 1)
```

last address in program flash

Definition at line 73 of file STM8AF_STM8S.h.

5.1.2.3458 STM8_PFLASH_SIZE [1/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8S208SB.h.

5.1.2.3459 STM8_PFLASH_SIZE [2/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8S207RB.h.

5.1.2.3460 STM8_PFLASH_SIZE [3/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8S207S6.h.

5.1.2.3461 STM8_PFLASH_SIZE [4/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8AF6286.h.

5.1.2.3462 STM8_PFLASH_SIZE [5/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8AF628A.h.

5.1.2.3463 STM8_PFLASH_SIZE [6/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8AF62AA.h.

5.1.2.3464 STM8_PFLASH_SIZE [7/71]

```
#define STM8_PFLASH_SIZE 16384
```

Definition at line 47 of file STM8AF6248.h.

5.1.2.3465 STM8_PFLASH_SIZE [8/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8S207MB.h.

5.1.2.3466 STM8_PFLASH_SIZE [9/71]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8S003F3.h.

5.1.2.3467 STM8_PFLASH_SIZE [10/71]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8S001J3.h.

5.1.2.3468 STM8_PFLASH_SIZE [11/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8AF6266.h.

5.1.2.3469 STM8_PFLASH_SIZE [12/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8S208RB.h.

5.1.2.3470 STM8_PFLASH_SIZE [13/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8AF52A9.h.

5.1.2.3471 STM8_PFLASH_SIZE [14/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8S207K6.h.

5.1.2.3472 STM8_PFLASH_SIZE [15/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8AF5289.h.

5.1.2.3473 STM8_PFLASH_SIZE [16/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8AF5286.h.

5.1.2.3474 STM8_PFLASH_SIZE [17/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8S207S8.h.

5.1.2.3475 STM8_PFLASH_SIZE [18/71]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8AF6223.h.

5.1.2.3476 STM8_PFLASH_SIZE [19/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8S208MB.h.

5.1.2.3477 STM8_PFLASH_SIZE [20/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8AF52A6.h.

5.1.2.3478 STM8_PFLASH_SIZE [21/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8AF5268.h.

5.1.2.3479 STM8_PFLASH_SIZE [22/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8S207K8.h.

5.1.2.3480 STM8_PFLASH_SIZE [23/71]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8S903F3.h.

5.1.2.3481 STM8_PFLASH_SIZE [24/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8S207R8.h.

5.1.2.3482 STM8_PFLASH_SIZE [25/71]

```
#define STM8_PFLASH_SIZE 4096
```

Definition at line 47 of file STM8AF6213.h.

5.1.2.3483 STM8_PFLASH_SIZE [26/71]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8S003K3.h.

5.1.2.3484 STM8_PFLASH_SIZE [27/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8S007C8.h.

5.1.2.3485 STM8_PFLASH_SIZE [28/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8S005C6.h.

5.1.2.3486 STM8_PFLASH_SIZE [29/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8AF6289.h.

5.1.2.3487 STM8_PFLASH_SIZE [30/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8S005K6.h.

5.1.2.3488 STM8_PFLASH_SIZE [31/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8AF6388.h.

5.1.2.3489 STM8_PFLASH_SIZE [32/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8S208C8.h.

5.1.2.3490 STM8_PFLASH_SIZE [33/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8S208S8.h.

5.1.2.3491 STM8_PFLASH_SIZE [34/71]

```
#define STM8_PFLASH_SIZE 4096
```

Definition at line 47 of file STM8S103F2.h.

5.1.2.3492 STM8_PFLASH_SIZE [35/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8AF62A9.h.

5.1.2.3493 STM8_PFLASH_SIZE [36/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8S207CB.h.

5.1.2.3494 STM8_PFLASH_SIZE [37/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8AF5288.h.

5.1.2.3495 STM8_PFLASH_SIZE [38/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8AF6268.h.

5.1.2.3496 STM8_PFLASH_SIZE [39/71]

```
#define STM8_PFLASH_SIZE 16384
```

Definition at line 47 of file STM8AF6246.h.

5.1.2.3497 STM8_PFLASH_SIZE [40/71]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8S103K3.h.

5.1.2.3498 STM8_PFLASH_SIZE [41/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8S208R6.h.

5.1.2.3499 STM8_PFLASH_SIZE [42/71]

```
#define STM8_PFLASH_SIZE 16384
```

Definition at line 47 of file STM8S105K4.h.

5.1.2.3500 STM8_PFLASH_SIZE [43/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8S208S6.h.

5.1.2.3501 STM8_PFLASH_SIZE [44/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8S208CB.h.

5.1.2.3502 STM8_PFLASH_SIZE [45/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8AF52AA.h.

5.1.2.3503 STM8_PFLASH_SIZE [46/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8S105K6.h.

5.1.2.3504 STM8_PFLASH_SIZE [47/71]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8AF6223A.h.

5.1.2.3505 STM8_PFLASH_SIZE [48/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8AF6269.h.

5.1.2.3506 STM8_PFLASH_SIZE [49/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8S208R8.h.

5.1.2.3507 STM8_PFLASH_SIZE [50/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8AF528A.h.

5.1.2.3508 STM8_PFLASH_SIZE [51/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8S207C6.h.

5.1.2.3509 STM8_PFLASH_SIZE [52/71]

```
#define STM8_PFLASH_SIZE 4096
```

Definition at line 47 of file STM8AF6213A.h.

5.1.2.3510 STM8_PFLASH_SIZE [53/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8S105S6.h.

5.1.2.3511 STM8_PFLASH_SIZE [54/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8S207C8.h.

5.1.2.3512 STM8_PFLASH_SIZE [55/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8AF5269.h.

5.1.2.3513 STM8_PFLASH_SIZE [56/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8AF62A8.h.

5.1.2.3514 STM8_PFLASH_SIZE [57/71]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8S903K3.h.

5.1.2.3515 STM8_PFLASH_SIZE [58/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8S208C6.h.

5.1.2.3516 STM8_PFLASH_SIZE [59/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8AF62A6.h.

5.1.2.3517 STM8_PFLASH_SIZE [60/71]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8S103F3.h.

5.1.2.3518 STM8_PFLASH_SIZE [61/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8S207R6.h.

5.1.2.3519 STM8_PFLASH_SIZE [62/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8AF6288.h.

5.1.2.3520 STM8_PFLASH_SIZE [63/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8S105C6.h.

5.1.2.3521 STM8_PFLASH_SIZE [64/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8S207SB.h.

5.1.2.3522 STM8_PFLASH_SIZE [65/71]

```
#define STM8_PFLASH_SIZE 32768
```

Definition at line 47 of file STM8AF6366.h.

5.1.2.3523 STM8_PFLASH_SIZE [66/71]

```
#define STM8_PFLASH_SIZE 16384
```

Definition at line 47 of file STM8S105S4.h.

5.1.2.3524 STM8_PFLASH_SIZE [67/71]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8AF6226.h.

5.1.2.3525 STM8_PFLASH_SIZE [68/71]

```
#define STM8_PFLASH_SIZE 65536
```

Definition at line 47 of file STM8S207M8.h.

5.1.2.3526 STM8_PFLASH_SIZE [69/71]

```
#define STM8_PFLASH_SIZE 131072
```

Definition at line 47 of file STM8AF52A8.h.

5.1.2.3527 STM8_PFLASH_SIZE [70/71]

```
#define STM8_PFLASH_SIZE 16384
```

Definition at line 47 of file STM8S105C4.h.

5.1.2.3528 STM8_PFLASH_SIZE [71/71]

```
#define STM8_PFLASH_SIZE 2048
```

size of program flash [B]

Definition at line 60 of file STM8AF_STM8S.h.

5.1.2.3529 STM8_PFLASH_START

```
#define STM8_PFLASH_START 0x8000
```

first address in program flash

Definition at line 72 of file STM8AF_STM8S.h.

5.1.2.3530 STM8_RAM_END

```
#define STM8_RAM_END (STM8_RAM_START + STM8_RAM_SIZE - 1)
```

last address in RAM

Definition at line 75 of file STM8AF_STM8S.h.

5.1.2.3531 STM8_RAM_SIZE [1/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207SB.h.

5.1.2.3532 STM8_RAM_SIZE [2/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF628A.h.

5.1.2.3533 STM8_RAM_SIZE [3/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8AF6266.h.

5.1.2.3534 STM8_RAM_SIZE [4/71]

```
#define STM8_RAM_SIZE 1024
```

Definition at line 48 of file STM8S903F3.h.

5.1.2.3535 STM8_RAM_SIZE [5/71]

```
#define STM8_RAM_SIZE 1024
```

Definition at line 48 of file STM8S103F2.h.

5.1.2.3536 STM8_RAM_SIZE [6/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S208SB.h.

5.1.2.3537 STM8_RAM_SIZE [7/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S007C8.h.

5.1.2.3538 STM8_RAM_SIZE [8/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF5288.h.

5.1.2.3539 STM8_RAM_SIZE [9/71]

```
#define STM8_RAM_SIZE 1024
```

Definition at line 48 of file STM8AF6213.h.

5.1.2.3540 STM8_RAM_SIZE [10/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207K8.h.

5.1.2.3541 STM8_RAM_SIZE [11/71]

```
#define STM8_RAM_SIZE 1024
```

Definition at line 48 of file STM8S001J3.h.

5.1.2.3542 STM8_RAM_SIZE [12/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207M8.h.

5.1.2.3543 STM8_RAM_SIZE [13/71]

```
#define STM8_RAM_SIZE 1024
```

Definition at line 48 of file STM8S003F3.h.

5.1.2.3544 STM8_RAM_SIZE [14/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF52A9.h.

5.1.2.3545 STM8_RAM_SIZE [15/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207K6.h.

5.1.2.3546 STM8_RAM_SIZE [16/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S208R6.h.

5.1.2.3547 STM8_RAM_SIZE [17/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8AF6268.h.

5.1.2.3548 STM8_RAM_SIZE [18/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF5286.h.

5.1.2.3549 STM8_RAM_SIZE [19/71]

```
#define STM8_RAM_SIZE 1024
```

Definition at line 48 of file STM8S103F3.h.

5.1.2.3550 STM8_RAM_SIZE [20/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8S105S6.h.

5.1.2.3551 STM8_RAM_SIZE [21/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207R6.h.

5.1.2.3552 STM8_RAM_SIZE [22/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8AF6246.h.

5.1.2.3553 STM8_RAM_SIZE [23/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S208S8.h.

5.1.2.3554 STM8_RAM_SIZE [24/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF62A8.h.

5.1.2.3555 STM8_RAM_SIZE [25/71]

```
#define STM8_RAM_SIZE 1024
```

Definition at line 48 of file STM8S103K3.h.

5.1.2.3556 STM8_RAM_SIZE [26/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF5289.h.

5.1.2.3557 STM8_RAM_SIZE [27/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8S005C6.h.

5.1.2.3558 STM8_RAM_SIZE [28/71]

```
#define STM8_RAM_SIZE 1024
```

Definition at line 48 of file STM8AF6223A.h.

5.1.2.3559 STM8_RAM_SIZE [29/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S208MB.h.

5.1.2.3560 STM8_RAM_SIZE [30/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8S105K6.h.

5.1.2.3561 STM8_RAM_SIZE [31/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S208RB.h.

5.1.2.3562 STM8_RAM_SIZE [32/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF528A.h.

5.1.2.3563 STM8_RAM_SIZE [33/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8S105C4.h.

5.1.2.3564 STM8_RAM_SIZE [34/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S208C6.h.

5.1.2.3565 STM8_RAM_SIZE [35/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207RB.h.

5.1.2.3566 STM8_RAM_SIZE [36/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8AF6226.h.

5.1.2.3567 STM8_RAM_SIZE [37/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF5268.h.

5.1.2.3568 STM8_RAM_SIZE [38/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF6388.h.

5.1.2.3569 STM8_RAM_SIZE [39/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207S8.h.

5.1.2.3570 STM8_RAM_SIZE [40/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF52AA.h.

5.1.2.3571 STM8_RAM_SIZE [41/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207S6.h.

5.1.2.3572 STM8_RAM_SIZE [42/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8AF6248.h.

5.1.2.3573 STM8_RAM_SIZE [43/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S208R8.h.

5.1.2.3574 STM8_RAM_SIZE [44/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF6289.h.

5.1.2.3575 STM8_RAM_SIZE [45/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207CB.h.

5.1.2.3576 STM8_RAM_SIZE [46/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF5269.h.

5.1.2.3577 STM8_RAM_SIZE [47/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S208C8.h.

5.1.2.3578 STM8_RAM_SIZE [48/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF52A8.h.

5.1.2.3579 STM8_RAM_SIZE [49/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF6269.h.

5.1.2.3580 STM8_RAM_SIZE [50/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF52A6.h.

5.1.2.3581 STM8_RAM_SIZE [51/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8AF6366.h.

5.1.2.3582 STM8_RAM_SIZE [52/71]

```
#define STM8_RAM_SIZE 1024
```

Definition at line 48 of file STM8S903K3.h.

5.1.2.3583 STM8_RAM_SIZE [53/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8S005K6.h.

5.1.2.3584 STM8_RAM_SIZE [54/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S208CB.h.

5.1.2.3585 STM8_RAM_SIZE [55/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF6288.h.

5.1.2.3586 STM8_RAM_SIZE [56/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF62AA.h.

5.1.2.3587 STM8_RAM_SIZE [57/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8S105S4.h.

5.1.2.3588 STM8_RAM_SIZE [58/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207MB.h.

5.1.2.3589 STM8_RAM_SIZE [59/71]

```
#define STM8_RAM_SIZE 1024
```

Definition at line 48 of file STM8S003K3.h.

5.1.2.3590 STM8_RAM_SIZE [60/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8S105C6.h.

5.1.2.3591 STM8_RAM_SIZE [61/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF62A6.h.

5.1.2.3592 STM8_RAM_SIZE [62/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF6286.h.

5.1.2.3593 STM8_RAM_SIZE [63/71]

```
#define STM8_RAM_SIZE 1024
```

Definition at line 48 of file STM8AF6213A.h.

5.1.2.3594 STM8_RAM_SIZE [64/71]

```
#define STM8_RAM_SIZE 2048
```

Definition at line 48 of file STM8S105K4.h.

5.1.2.3595 STM8_RAM_SIZE [65/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207C8.h.

5.1.2.3596 STM8_RAM_SIZE [66/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S208S6.h.

5.1.2.3597 STM8_RAM_SIZE [67/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207C6.h.

5.1.2.3598 STM8_RAM_SIZE [68/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8AF62A9.h.

5.1.2.3599 STM8_RAM_SIZE [69/71]

```
#define STM8_RAM_SIZE 1024
```

Definition at line 48 of file STM8AF6223.h.

5.1.2.3600 STM8_RAM_SIZE [70/71]

```
#define STM8_RAM_SIZE 6144
```

Definition at line 48 of file STM8S207R8.h.

5.1.2.3601 STM8_RAM_SIZE [71/71]

```
#define STM8_RAM_SIZE 1024
```

size of RAM [B]

Definition at line 64 of file STM8AF_STM8S.h.

5.1.2.3602 STM8_RAM_START

```
#define STM8_RAM_START 0x0000
```

first address in RAM

Definition at line 74 of file STM8AF_STM8S.h.

5.1.2.3603 STM8AF5268

```
#define STM8AF5268
```

Definition at line 40 of file STM8AF5268.h.

5.1.2.3604 STM8AF5269

```
#define STM8AF5269
```

Definition at line 40 of file STM8AF5269.h.

5.1.2.3605 STM8AF526x [1/2]

```
#define STM8AF526x
```

Definition at line 43 of file STM8AF5269.h.

5.1.2.3606 STM8AF526x [2/2]

```
#define STM8AF526x
```

Definition at line 43 of file STM8AF5268.h.

5.1.2.3607 STM8AF5286

```
#define STM8AF5286
```

Definition at line 40 of file STM8AF5286.h.

5.1.2.3608 STM8AF5288

```
#define STM8AF5288
```

Definition at line 40 of file STM8AF5288.h.

5.1.2.3609 STM8AF5289

```
#define STM8AF5289
```

Definition at line 40 of file STM8AF5289.h.

5.1.2.3610 STM8AF528A

```
#define STM8AF528A
```

Definition at line 40 of file STM8AF528A.h.

5.1.2.3611 STM8AF528x [1/4]

```
#define STM8AF528x
```

Definition at line 43 of file STM8AF528A.h.

5.1.2.3612 STM8AF528x [2/4]

```
#define STM8AF528x
```

Definition at line 43 of file STM8AF5286.h.

5.1.2.3613 STM8AF528x [3/4]

```
#define STM8AF528x
```

Definition at line 43 of file STM8AF5289.h.

5.1.2.3614 STM8AF528x [4/4]

```
#define STM8AF528x
```

Definition at line 43 of file STM8AF5288.h.

5.1.2.3615 STM8AF52A6

```
#define STM8AF52A6
```

Definition at line 40 of file STM8AF52A6.h.

5.1.2.3616 STM8AF52A8

```
#define STM8AF52A8
```

Definition at line 40 of file STM8AF52A8.h.

5.1.2.3617 STM8AF52A9

```
#define STM8AF52A9
```

Definition at line 40 of file STM8AF52A9.h.

5.1.2.3618 STM8AF52AA

```
#define STM8AF52AA
```

Definition at line 40 of file STM8AF52AA.h.

5.1.2.3619 STM8AF52Ax [1/4]

```
#define STM8AF52Ax
```

Definition at line 43 of file STM8AF52A9.h.

5.1.2.3620 STM8AF52Ax [2/4]

```
#define STM8AF52Ax
```

Definition at line 43 of file STM8AF52AA.h.

5.1.2.3621 STM8AF52Ax [3/4]

```
#define STM8AF52Ax
```

Definition at line 43 of file STM8AF52A8.h.

5.1.2.3622 STM8AF52Ax [4/4]

```
#define STM8AF52Ax
```

Definition at line 43 of file STM8AF52A6.h.

5.1.2.3623 STM8AF6213

```
#define STM8AF6213
```

Definition at line 40 of file STM8AF6213.h.

5.1.2.3624 STM8AF6213A

```
#define STM8AF6213A
```

Definition at line 40 of file STM8AF6213A.h.

5.1.2.3625 STM8AF621x [1/2]

```
#define STM8AF621x
```

Definition at line 43 of file STM8AF6213.h.

5.1.2.3626 STM8AF621x [2/2]

```
#define STM8AF621x
```

Definition at line 43 of file STM8AF6213A.h.

5.1.2.3627 STM8AF6223

```
#define STM8AF6223
```

Definition at line 40 of file STM8AF6223.h.

5.1.2.3628 STM8AF6223A

```
#define STM8AF6223A
```

Definition at line 40 of file STM8AF6223A.h.

5.1.2.3629 STM8AF6226

```
#define STM8AF6226
```

Definition at line 40 of file STM8AF6226.h.

5.1.2.3630 STM8AF622x [1/3]

```
#define STM8AF622x
```

Definition at line 43 of file STM8AF6223A.h.

5.1.2.3631 STM8AF622x [2/3]

```
#define STM8AF622x
```

Definition at line 43 of file STM8AF6223.h.

5.1.2.3632 STM8AF622x [3/3]

```
#define STM8AF622x
```

Definition at line 43 of file STM8AF6226.h.

5.1.2.3633 STM8AF6246

```
#define STM8AF6246
```

Definition at line 40 of file STM8AF6246.h.

5.1.2.3634 STM8AF6248

```
#define STM8AF6248
```

Definition at line 40 of file STM8AF6248.h.

5.1.2.3635 STM8AF624x [1/2]

```
#define STM8AF624x
```

Definition at line 43 of file STM8AF6246.h.

5.1.2.3636 STM8AF624x [2/2]

```
#define STM8AF624x
```

Definition at line 43 of file STM8AF6248.h.

5.1.2.3637 STM8AF6266

```
#define STM8AF6266
```

Definition at line 40 of file STM8AF6266.h.

5.1.2.3638 STM8AF6268

```
#define STM8AF6268
```

Definition at line 40 of file STM8AF6268.h.

5.1.2.3639 STM8AF6269

```
#define STM8AF6269
```

Definition at line 40 of file STM8AF6269.h.

5.1.2.3640 STM8AF626x [1/3]

```
#define STM8AF626x
```

Definition at line 43 of file STM8AF6266.h.

5.1.2.3641 STM8AF626x [2/3]

```
#define STM8AF626x
```

Definition at line 43 of file STM8AF6268.h.

5.1.2.3642 STM8AF626x [3/3]

```
#define STM8AF626x
```

Definition at line 43 of file STM8AF6269.h.

5.1.2.3643 STM8AF6286

```
#define STM8AF6286
```

Definition at line 40 of file STM8AF6286.h.

5.1.2.3644 STM8AF6288

```
#define STM8AF6288
```

Definition at line 40 of file STM8AF6288.h.

5.1.2.3645 STM8AF6289

```
#define STM8AF6289
```

Definition at line 40 of file STM8AF6289.h.

5.1.2.3646 STM8AF628A

```
#define STM8AF628A
```

Definition at line 40 of file STM8AF628A.h.

5.1.2.3647 STM8AF628x [1/4]

```
#define STM8AF628x
```

Definition at line 43 of file STM8AF6288.h.

5.1.2.3648 STM8AF628x [2/4]

```
#define STM8AF628x
```

Definition at line 43 of file STM8AF6289.h.

5.1.2.3649 STM8AF628x [3/4]

```
#define STM8AF628x
```

Definition at line 43 of file STM8AF6286.h.

5.1.2.3650 STM8AF628x [4/4]

```
#define STM8AF628x
```

Definition at line 43 of file STM8AF628A.h.

5.1.2.3651 STM8AF62A6

```
#define STM8AF62A6
```

Definition at line 40 of file STM8AF62A6.h.

5.1.2.3652 STM8AF62A8

```
#define STM8AF62A8
```

Definition at line 40 of file STM8AF62A8.h.

5.1.2.3653 STM8AF62A9

```
#define STM8AF62A9
```

Definition at line 40 of file STM8AF62A9.h.

5.1.2.3654 STM8AF62AA

```
#define STM8AF62AA
```

Definition at line 40 of file STM8AF62AA.h.

5.1.2.3655 STM8AF62Ax [1/4]

```
#define STM8AF62Ax
```

Definition at line 43 of file STM8AF62AA.h.

5.1.2.3656 STM8AF62Ax [2/4]

```
#define STM8AF62Ax
```

Definition at line 43 of file STM8AF62A9.h.

5.1.2.3657 STM8AF62Ax [3/4]

```
#define STM8AF62Ax
```

Definition at line 43 of file STM8AF62A6.h.

5.1.2.3658 STM8AF62Ax [4/4]

```
#define STM8AF62Ax
```

Definition at line 43 of file STM8AF62A8.h.

5.1.2.3659 STM8AF6366

```
#define STM8AF6366
```

Definition at line 40 of file STM8AF6366.h.

5.1.2.3660 STM8AF636x

```
#define STM8AF636x
```

Definition at line 43 of file STM8AF6366.h.

5.1.2.3661 STM8AF6388

```
#define STM8AF6388
```

Definition at line 40 of file STM8AF6388.h.

5.1.2.3662 STM8AF638x

```
#define STM8AF638x
```

Definition at line 43 of file STM8AF6388.h.

5.1.2.3663 STM8S001

```
#define STM8S001
```

Definition at line 43 of file STM8S001J3.h.

5.1.2.3664 STM8S001J3

```
#define STM8S001J3
```

Definition at line 40 of file STM8S001J3.h.

5.1.2.3665 STM8S003 [1/2]

```
#define STM8S003
```

Definition at line 43 of file STM8S003F3.h.

5.1.2.3666 STM8S003 [2/2]

```
#define STM8S003
```

Definition at line 43 of file STM8S003K3.h.

5.1.2.3667 STM8S003F3

```
#define STM8S003F3
```

Definition at line 40 of file STM8S003F3.h.

5.1.2.3668 STM8S003K3

```
#define STM8S003K3
```

Definition at line 40 of file STM8S003K3.h.

5.1.2.3669 STM8S005 [1/2]

```
#define STM8S005
```

Definition at line 43 of file STM8S005K6.h.

5.1.2.3670 STM8S005 [2/2]

```
#define STM8S005
```

Definition at line 43 of file STM8S005C6.h.

5.1.2.3671 STM8S005C6

```
#define STM8S005C6
```

Definition at line 40 of file STM8S005C6.h.

5.1.2.3672 STM8S005K6

```
#define STM8S005K6
```

Definition at line 40 of file STM8S005K6.h.

5.1.2.3673 STM8S007

```
#define STM8S007
```

Definition at line 43 of file STM8S007C8.h.

5.1.2.3674 STM8S007C8

```
#define STM8S007C8
```

Definition at line 40 of file STM8S007C8.h.

5.1.2.3675 STM8S103 [1/3]

```
#define STM8S103
```

Definition at line 43 of file STM8S103F2.h.

5.1.2.3676 STM8S103 [2/3]

```
#define STM8S103
```

Definition at line 43 of file STM8S103F3.h.

5.1.2.3677 STM8S103 [3/3]

```
#define STM8S103
```

Definition at line 43 of file STM8S103K3.h.

5.1.2.3678 STM8S103F2

```
#define STM8S103F2
```

Definition at line 40 of file STM8S103F2.h.

5.1.2.3679 STM8S103F3

```
#define STM8S103F3
```

Definition at line 40 of file STM8S103F3.h.

5.1.2.3680 STM8S103K3

```
#define STM8S103K3
```

Definition at line 40 of file STM8S103K3.h.

5.1.2.3681 STM8S105 [1/6]

```
#define STM8S105
```

Definition at line 43 of file STM8S105K4.h.

5.1.2.3682 STM8S105 [2/6]

```
#define STM8S105
```

Definition at line 43 of file STM8S105C6.h.

5.1.2.3683 STM8S105 [3/6]

```
#define STM8S105
```

Definition at line 43 of file STM8S105C4.h.

5.1.2.3684 STM8S105 [4/6]

```
#define STM8S105
```

Definition at line 43 of file STM8S105S4.h.

5.1.2.3685 STM8S105 [5/6]

```
#define STM8S105
```

Definition at line 43 of file STM8S105S6.h.

5.1.2.3686 STM8S105 [6/6]

```
#define STM8S105
```

Definition at line 43 of file STM8S105K6.h.

5.1.2.3687 STM8S105C4

```
#define STM8S105C4
```

Definition at line 40 of file STM8S105C4.h.

5.1.2.3688 STM8S105C6

```
#define STM8S105C6
```

Definition at line 40 of file STM8S105C6.h.

5.1.2.3689 STM8S105K4

```
#define STM8S105K4
```

Definition at line 40 of file STM8S105K4.h.

5.1.2.3690 STM8S105K6

```
#define STM8S105K6
```

Definition at line 40 of file STM8S105K6.h.

5.1.2.3691 STM8S105S4

```
#define STM8S105S4
```

Definition at line 40 of file STM8S105S4.h.

5.1.2.3692 STM8S105S6

```
#define STM8S105S6
```

Definition at line 40 of file STM8S105S6.h.

5.1.2.3693 STM8S207 [1/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207S6.h.

5.1.2.3694 STM8S207 [2/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207CB.h.

5.1.2.3695 STM8S207 [3/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207R6.h.

5.1.2.3696 STM8S207 [4/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207M8.h.

5.1.2.3697 STM8S207 [5/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207K6.h.

5.1.2.3698 STM8S207 [6/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207R8.h.

5.1.2.3699 STM8S207 [7/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207S8.h.

5.1.2.3700 STM8S207 [8/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207RB.h.

5.1.2.3701 STM8S207 [9/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207SB.h.

5.1.2.3702 STM8S207 [10/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207MB.h.

5.1.2.3703 STM8S207 [11/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207C8.h.

5.1.2.3704 STM8S207 [12/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207C6.h.

5.1.2.3705 STM8S207 [13/13]

```
#define STM8S207
```

Definition at line 43 of file STM8S207K8.h.

5.1.2.3706 STM8S207C6

```
#define STM8S207C6
```

Definition at line 40 of file STM8S207C6.h.

5.1.2.3707 STM8S207C8

```
#define STM8S207C8
```

Definition at line 40 of file STM8S207C8.h.

5.1.2.3708 STM8S207CB

```
#define STM8S207CB
```

Definition at line 40 of file STM8S207CB.h.

5.1.2.3709 STM8S207K6

```
#define STM8S207K6
```

Definition at line 40 of file STM8S207K6.h.

5.1.2.3710 STM8S207K8

```
#define STM8S207K8
```

Definition at line 40 of file STM8S207K8.h.

5.1.2.3711 STM8S207M8

```
#define STM8S207M8
```

Definition at line 40 of file STM8S207M8.h.

5.1.2.3712 STM8S207MB

```
#define STM8S207MB
```

Definition at line 40 of file STM8S207MB.h.

5.1.2.3713 STM8S207R6

```
#define STM8S207R6
```

Definition at line 40 of file STM8S207R6.h.

5.1.2.3714 STM8S207R8

```
#define STM8S207R8
```

Definition at line 40 of file STM8S207R8.h.

5.1.2.3715 STM8S207RB

```
#define STM8S207RB
```

Definition at line 40 of file STM8S207RB.h.

5.1.2.3716 STM8S207S6

```
#define STM8S207S6
```

Definition at line 40 of file STM8S207S6.h.

5.1.2.3717 STM8S207S8

```
#define STM8S207S8
```

Definition at line 40 of file STM8S207S8.h.

5.1.2.3718 STM8S207SB

```
#define STM8S207SB
```

Definition at line 40 of file STM8S207SB.h.

5.1.2.3719 STM8S208 [1/10]

```
#define STM8S208
```

Definition at line 43 of file STM8S208R8.h.

5.1.2.3720 STM8S208 [2/10]

```
#define STM8S208
```

Definition at line 43 of file STM8S208C8.h.

5.1.2.3721 STM8S208 [3/10]

```
#define STM8S208
```

Definition at line 43 of file STM8S208CB.h.

5.1.2.3722 STM8S208 [4/10]

```
#define STM8S208
```

Definition at line 43 of file STM8S208R6.h.

5.1.2.3723 STM8S208 [5/10]

```
#define STM8S208
```

Definition at line 43 of file STM8S208C6.h.

5.1.2.3724 STM8S208 [6/10]

```
#define STM8S208
```

Definition at line 43 of file STM8S208RB.h.

5.1.2.3725 STM8S208 [7/10]

```
#define STM8S208
```

Definition at line 43 of file STM8S208S8.h.

5.1.2.3726 STM8S208 [8/10]

```
#define STM8S208
```

Definition at line 43 of file STM8S208SB.h.

5.1.2.3727 STM8S208 [9/10]

```
#define STM8S208
```

Definition at line 43 of file STM8S208MB.h.

5.1.2.3728 STM8S208 [10/10]

```
#define STM8S208
```

Definition at line 43 of file STM8S208S6.h.

5.1.2.3729 STM8S208C6

```
#define STM8S208C6
```

Definition at line 40 of file STM8S208C6.h.

5.1.2.3730 STM8S208C8

```
#define STM8S208C8
```

Definition at line 40 of file STM8S208C8.h.

5.1.2.3731 STM8S208CB

```
#define STM8S208CB
```

Definition at line 40 of file STM8S208CB.h.

5.1.2.3732 STM8S208MB

```
#define STM8S208MB
```

Definition at line 40 of file STM8S208MB.h.

5.1.2.3733 STM8S208R6

```
#define STM8S208R6
```

Definition at line 40 of file STM8S208R6.h.

5.1.2.3734 STM8S208R8

```
#define STM8S208R8
```

Definition at line 40 of file STM8S208R8.h.

5.1.2.3735 STM8S208RB

```
#define STM8S208RB
```

Definition at line 40 of file STM8S208RB.h.

5.1.2.3736 STM8S208S6

```
#define STM8S208S6
```

Definition at line 40 of file STM8S208S6.h.

5.1.2.3737 STM8S208S8

```
#define STM8S208S8
```

Definition at line 40 of file STM8S208S8.h.

5.1.2.3738 STM8S208SB

```
#define STM8S208SB
```

Definition at line 40 of file STM8S208SB.h.

5.1.2.3739 STM8S903 [1/2]

```
#define STM8S903
```

Definition at line 43 of file STM8S903K3.h.

5.1.2.3740 STM8S903 [2/2]

```
#define STM8S903
```

Definition at line 43 of file STM8S903F3.h.

5.1.2.3741 STM8S903F3

```
#define STM8S903F3
```

Definition at line 40 of file STM8S903F3.h.

5.1.2.3742 STM8S903K3

```
#define STM8S903K3
```

Definition at line 40 of file STM8S903K3.h.

5.1.2.3743 SW_RESET

```
#define SW_RESET( ) (_WWDG_CR=0xBF)
```

reset controller via WWGD module

Definition at line 174 of file STM8AF_STM8S.h.

5.1.2.3744 TIM1_AddressBase [1/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8S103K3.h.

5.1.2.3745 TIM1_AddressBase [2/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8AF6226.h.

5.1.2.3746 TIM1_AddressBase [3/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8S001J3.h.

5.1.2.3747 TIM1_AddressBase [4/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8S003K3.h.

5.1.2.3748 TIM1_AddressBase [5/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8AF6223.h.

5.1.2.3749 TIM1_AddressBase [6/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8AF6213.h.

5.1.2.3750 TIM1_AddressBase [7/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8S103F2.h.

5.1.2.3751 TIM1_AddressBase [8/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8S903F3.h.

5.1.2.3752 TIM1_AddressBase [9/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8AF6366.h.

5.1.2.3753 TIM1_AddressBase [10/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8S003F3.h.

5.1.2.3754 TIM1_AddressBase [11/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8AF6223A.h.

5.1.2.3755 TIM1_AddressBase [12/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8S903K3.h.

5.1.2.3756 TIM1_AddressBase [13/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8AF6213A.h.

5.1.2.3757 TIM1_AddressBase [14/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 70 of file STM8S103F3.h.

5.1.2.3758 TIM1_AddressBase [15/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8S105S6.h.

5.1.2.3759 TIM1_AddressBase [16/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8S005K6.h.

5.1.2.3760 TIM1_AddressBase [17/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8AF6246.h.

5.1.2.3761 TIM1_AddressBase [18/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8S105C6.h.

5.1.2.3762 TIM1_AddressBase [19/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8AF6269.h.

5.1.2.3763 TIM1_AddressBase [20/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8S005C6.h.

5.1.2.3764 TIM1_AddressBase [21/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8AF6248.h.

5.1.2.3765 TIM1_AddressBase [22/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8AF6268.h.

5.1.2.3766 TIM1_AddressBase [23/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8S105S4.h.

5.1.2.3767 TIM1_AddressBase [24/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8AF6266.h.

5.1.2.3768 TIM1_AddressBase [25/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8S105K4.h.

5.1.2.3769 TIM1_AddressBase [26/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8S105K6.h.

5.1.2.3770 TIM1_AddressBase [27/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 71 of file STM8S105C4.h.

5.1.2.3771 TIM1_AddressBase [28/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF628A.h.

5.1.2.3772 TIM1_AddressBase [29/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF6286.h.

5.1.2.3773 TIM1_AddressBase [30/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF62A9.h.

5.1.2.3774 TIM1_AddressBase [31/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207C6.h.

5.1.2.3775 TIM1_AddressBase [32/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207MB.h.

5.1.2.3776 TIM1_AddressBase [33/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S208C8.h.

5.1.2.3777 TIM1_AddressBase [34/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207R6.h.

5.1.2.3778 TIM1_AddressBase [35/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S208SB.h.

5.1.2.3779 TIM1_AddressBase [36/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF6289.h.

5.1.2.3780 TIM1_AddressBase [37/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207CB.h.

5.1.2.3781 TIM1_AddressBase [38/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF52AA.h.

5.1.2.3782 TIM1_AddressBase [39/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S208C6.h.

5.1.2.3783 TIM1_AddressBase [40/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207M8.h.

5.1.2.3784 TIM1_AddressBase [41/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF5268.h.

5.1.2.3785 TIM1_AddressBase [42/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207RB.h.

5.1.2.3786 TIM1_AddressBase [43/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF62A8.h.

5.1.2.3787 TIM1_AddressBase [44/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF5289.h.

5.1.2.3788 TIM1_AddressBase [45/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S208RB.h.

5.1.2.3789 TIM1_AddressBase [46/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF52A6.h.

5.1.2.3790 TIM1_AddressBase [47/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S208MB.h.

5.1.2.3791 TIM1_AddressBase [48/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207R8.h.

5.1.2.3792 TIM1_AddressBase [49/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF5288.h.

5.1.2.3793 TIM1_AddressBase [50/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S208R6.h.

5.1.2.3794 TIM1_AddressBase [51/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207C8.h.

5.1.2.3795 TIM1_AddressBase [52/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S208CB.h.

5.1.2.3796 TIM1_AddressBase [53/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207K6.h.

5.1.2.3797 TIM1_AddressBase [54/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207S6.h.

5.1.2.3798 TIM1_AddressBase [55/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF62AA.h.

5.1.2.3799 TIM1_AddressBase [56/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S208S6.h.

5.1.2.3800 TIM1_AddressBase [57/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF52A9.h.

5.1.2.3801 TIM1_AddressBase [58/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S208R8.h.

5.1.2.3802 TIM1_AddressBase [59/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S007C8.h.

5.1.2.3803 TIM1_AddressBase [60/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207S8.h.

5.1.2.3804 TIM1_AddressBase [61/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S208S8.h.

5.1.2.3805 TIM1_AddressBase [62/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF6288.h.

5.1.2.3806 TIM1_AddressBase [63/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF5269.h.

5.1.2.3807 TIM1_AddressBase [64/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207SB.h.

5.1.2.3808 TIM1_AddressBase [65/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8S207K8.h.

5.1.2.3809 TIM1_AddressBase [66/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF6388.h.

5.1.2.3810 TIM1_AddressBase [67/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF528A.h.

5.1.2.3811 TIM1_AddressBase [68/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF52A8.h.

5.1.2.3812 TIM1_AddressBase [69/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF62A6.h.

5.1.2.3813 TIM1_AddressBase [70/70]

```
#define TIM1_AddressBase 0x5250
```

Definition at line 74 of file STM8AF5286.h.

5.1.2.3814 TIM2_AddressBase [1/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 71 of file STM8S001J3.h.

5.1.2.3815 TIM2_AddressBase [2/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 71 of file STM8S003F3.h.

5.1.2.3816 TIM2_AddressBase [3/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 71 of file STM8S103F3.h.

5.1.2.3817 TIM2_AddressBase [4/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 71 of file STM8AF6366.h.

5.1.2.3818 TIM2_AddressBase [5/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 71 of file STM8S103F2.h.

5.1.2.3819 TIM2_AddressBase [6/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 71 of file STM8S003K3.h.

5.1.2.3820 TIM2_AddressBase [7/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 71 of file STM8S103K3.h.

5.1.2.3821 TIM2_AddressBase [8/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8S005C6.h.

5.1.2.3822 TIM2_AddressBase [9/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8S005K6.h.

5.1.2.3823 TIM2_AddressBase [10/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8S105K6.h.

5.1.2.3824 TIM2_AddressBase [11/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8AF6266.h.

5.1.2.3825 TIM2_AddressBase [12/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8AF6246.h.

5.1.2.3826 TIM2_AddressBase [13/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8S105S4.h.

5.1.2.3827 TIM2_AddressBase [14/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8S105S6.h.

5.1.2.3828 TIM2_AddressBase [15/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8AF6269.h.

5.1.2.3829 TIM2_AddressBase [16/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8S105C6.h.

5.1.2.3830 TIM2_AddressBase [17/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8AF6268.h.

5.1.2.3831 TIM2_AddressBase [18/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8S105C4.h.

5.1.2.3832 TIM2_AddressBase [19/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8AF6248.h.

5.1.2.3833 TIM2_AddressBase [20/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 72 of file STM8S105K4.h.

5.1.2.3834 TIM2_AddressBase [21/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S208MB.h.

5.1.2.3835 TIM2_AddressBase [22/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF5289.h.

5.1.2.3836 TIM2_AddressBase [23/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF6288.h.

5.1.2.3837 TIM2_AddressBase [24/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207S6.h.

5.1.2.3838 TIM2_AddressBase [25/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207SB.h.

5.1.2.3839 TIM2_AddressBase [26/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207K8.h.

5.1.2.3840 TIM2_AddressBase [27/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF52A8.h.

5.1.2.3841 TIM2_AddressBase [28/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S208R8.h.

5.1.2.3842 TIM2_AddressBase [29/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S208CB.h.

5.1.2.3843 TIM2_AddressBase [30/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S208SB.h.

5.1.2.3844 TIM2_AddressBase [31/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF5269.h.

5.1.2.3845 TIM2_AddressBase [32/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF628A.h.

5.1.2.3846 TIM2_AddressBase [33/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF52A9.h.

5.1.2.3847 TIM2_AddressBase [34/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF6289.h.

5.1.2.3848 TIM2_AddressBase [35/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF6286.h.

5.1.2.3849 TIM2_AddressBase [36/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF52AA.h.

5.1.2.3850 TIM2_AddressBase [37/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207M8.h.

5.1.2.3851 TIM2_AddressBase [38/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207MB.h.

5.1.2.3852 TIM2_AddressBase [39/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207R8.h.

5.1.2.3853 TIM2_AddressBase [40/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF6388.h.

5.1.2.3854 TIM2_AddressBase [41/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207K6.h.

5.1.2.3855 TIM2_AddressBase [42/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF62AA.h.

5.1.2.3856 TIM2_AddressBase [43/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207C6.h.

5.1.2.3857 TIM2_AddressBase [44/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207R6.h.

5.1.2.3858 TIM2_AddressBase [45/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF62A9.h.

5.1.2.3859 TIM2_AddressBase [46/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF5286.h.

5.1.2.3860 TIM2_AddressBase [47/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S208C8.h.

5.1.2.3861 TIM2_AddressBase [48/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF528A.h.

5.1.2.3862 TIM2_AddressBase [49/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S208R6.h.

5.1.2.3863 TIM2_AddressBase [50/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF5268.h.

5.1.2.3864 TIM2_AddressBase [51/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S208RB.h.

5.1.2.3865 TIM2_AddressBase [52/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF52A6.h.

5.1.2.3866 TIM2_AddressBase [53/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S208S6.h.

5.1.2.3867 TIM2_AddressBase [54/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF62A8.h.

5.1.2.3868 TIM2_AddressBase [55/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207RB.h.

5.1.2.3869 TIM2_AddressBase [56/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S007C8.h.

5.1.2.3870 TIM2_AddressBase [57/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S208S8.h.

5.1.2.3871 TIM2_AddressBase [58/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S208C6.h.

5.1.2.3872 TIM2_AddressBase [59/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207C8.h.

5.1.2.3873 TIM2_AddressBase [60/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF62A6.h.

5.1.2.3874 TIM2_AddressBase [61/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207S8.h.

5.1.2.3875 TIM2_AddressBase [62/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8AF5288.h.

5.1.2.3876 TIM2_AddressBase [63/63]

```
#define TIM2_AddressBase 0x5300
```

Definition at line 75 of file STM8S207CB.h.

5.1.2.3877 TIM3_AddressBase [1/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 72 of file STM8AF6366.h.

5.1.2.3878 TIM3_AddressBase [2/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8AF6269.h.

5.1.2.3879 TIM3_AddressBase [3/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8S005K6.h.

5.1.2.3880 TIM3_AddressBase [4/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8S105C6.h.

5.1.2.3881 TIM3_AddressBase [5/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8AF6246.h.

5.1.2.3882 TIM3_AddressBase [6/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8AF6268.h.

5.1.2.3883 TIM3_AddressBase [7/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8S105K4.h.

5.1.2.3884 TIM3_AddressBase [8/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8S105S4.h.

5.1.2.3885 TIM3_AddressBase [9/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8S105C4.h.

5.1.2.3886 TIM3_AddressBase [10/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8S105S6.h.

5.1.2.3887 TIM3_AddressBase [11/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8AF6248.h.

5.1.2.3888 TIM3_AddressBase [12/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8AF6266.h.

5.1.2.3889 TIM3_AddressBase [13/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8S005C6.h.

5.1.2.3890 TIM3_AddressBase [14/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 73 of file STM8S105K6.h.

5.1.2.3891 TIM3_AddressBase [15/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF628A.h.

5.1.2.3892 TIM3_AddressBase [16/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF6286.h.

5.1.2.3893 TIM3_AddressBase [17/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF62A8.h.

5.1.2.3894 TIM3_AddressBase [18/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S007C8.h.

5.1.2.3895 TIM3_AddressBase [19/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S208C6.h.

5.1.2.3896 TIM3_AddressBase [20/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S208C8.h.

5.1.2.3897 TIM3_AddressBase [21/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207MB.h.

5.1.2.3898 TIM3_AddressBase [22/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF5289.h.

5.1.2.3899 TIM3_AddressBase [23/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207C8.h.

5.1.2.3900 TIM3_AddressBase [24/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S208MB.h.

5.1.2.3901 TIM3_AddressBase [25/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF5286.h.

5.1.2.3902 TIM3_AddressBase [26/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF5269.h.

5.1.2.3903 TIM3_AddressBase [27/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207SB.h.

5.1.2.3904 TIM3_AddressBase [28/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF52A6.h.

5.1.2.3905 TIM3_AddressBase [29/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S208S8.h.

5.1.2.3906 TIM3_AddressBase [30/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207RB.h.

5.1.2.3907 TIM3_AddressBase [31/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207R6.h.

5.1.2.3908 TIM3_AddressBase [32/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF52A8.h.

5.1.2.3909 TIM3_AddressBase [33/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207K6.h.

5.1.2.3910 TIM3_AddressBase [34/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207M8.h.

5.1.2.3911 TIM3_AddressBase [35/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S208SB.h.

5.1.2.3912 TIM3_AddressBase [36/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF5268.h.

5.1.2.3913 TIM3_AddressBase [37/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S208R8.h.

5.1.2.3914 TIM3_AddressBase [38/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S208S6.h.

5.1.2.3915 TIM3_AddressBase [39/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207K8.h.

5.1.2.3916 TIM3_AddressBase [40/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF5288.h.

5.1.2.3917 TIM3_AddressBase [41/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207C6.h.

5.1.2.3918 TIM3_AddressBase [42/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S208CB.h.

5.1.2.3919 TIM3_AddressBase [43/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S208R6.h.

5.1.2.3920 TIM3_AddressBase [44/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207CB.h.

5.1.2.3921 TIM3_AddressBase [45/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207S8.h.

5.1.2.3922 TIM3_AddressBase [46/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF62AA.h.

5.1.2.3923 TIM3_AddressBase [47/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF62A9.h.

5.1.2.3924 TIM3_AddressBase [48/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF6388.h.

5.1.2.3925 TIM3_AddressBase [49/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF6288.h.

5.1.2.3926 TIM3_AddressBase [50/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S208RB.h.

5.1.2.3927 TIM3_AddressBase [51/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207R8.h.

5.1.2.3928 TIM3_AddressBase [52/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8S207S6.h.

5.1.2.3929 TIM3_AddressBase [53/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF52AA.h.

5.1.2.3930 TIM3_AddressBase [54/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF528A.h.

5.1.2.3931 TIM3_AddressBase [55/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF52A9.h.

5.1.2.3932 TIM3_AddressBase [56/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF62A6.h.

5.1.2.3933 TIM3_AddressBase [57/57]

```
#define TIM3_AddressBase 0x5320
```

Definition at line 76 of file STM8AF6289.h.

5.1.2.3934 TIM4_AddressBase [1/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 72 of file STM8S003F3.h.

5.1.2.3935 TIM4_AddressBase [2/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 72 of file STM8S103K3.h.

5.1.2.3936 TIM4_AddressBase [3/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 72 of file STM8S003K3.h.

5.1.2.3937 TIM4_AddressBase [4/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 72 of file STM8S103F2.h.

5.1.2.3938 TIM4_AddressBase [5/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 72 of file STM8S001J3.h.

5.1.2.3939 TIM4_AddressBase [6/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 72 of file STM8S103F3.h.

5.1.2.3940 TIM4_AddressBase [7/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 73 of file STM8AF6366.h.

5.1.2.3941 TIM4_AddressBase [8/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8S005C6.h.

5.1.2.3942 TIM4_AddressBase [9/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8AF6268.h.

5.1.2.3943 TIM4_AddressBase [10/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8AF6266.h.

5.1.2.3944 TIM4_AddressBase [11/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8S105K4.h.

5.1.2.3945 TIM4_AddressBase [12/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8AF6269.h.

5.1.2.3946 TIM4_AddressBase [13/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8S105S6.h.

5.1.2.3947 TIM4_AddressBase [14/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8S005K6.h.

5.1.2.3948 TIM4_AddressBase [15/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8S105K6.h.

5.1.2.3949 TIM4_AddressBase [16/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8S105C6.h.

5.1.2.3950 TIM4_AddressBase [17/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8AF6248.h.

5.1.2.3951 TIM4_AddressBase [18/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8AF6246.h.

5.1.2.3952 TIM4_AddressBase [19/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8S105C4.h.

5.1.2.3953 TIM4_AddressBase [20/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 74 of file STM8S105S4.h.

5.1.2.3954 TIM4_AddressBase [21/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207SB.h.

5.1.2.3955 TIM4_AddressBase [22/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S208CB.h.

5.1.2.3956 TIM4_AddressBase [23/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF62AA.h.

5.1.2.3957 TIM4_AddressBase [24/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207S6.h.

5.1.2.3958 TIM4_AddressBase [25/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF62A9.h.

5.1.2.3959 TIM4_AddressBase [26/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF5286.h.

5.1.2.3960 TIM4_AddressBase [27/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207K6.h.

5.1.2.3961 TIM4_AddressBase [28/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207R8.h.

5.1.2.3962 TIM4_AddressBase [29/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S208MB.h.

5.1.2.3963 TIM4_AddressBase [30/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF528A.h.

5.1.2.3964 TIM4_AddressBase [31/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207RB.h.

5.1.2.3965 TIM4_AddressBase [32/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF52A8.h.

5.1.2.3966 TIM4_AddressBase [33/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S208C8.h.

5.1.2.3967 TIM4_AddressBase [34/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207CB.h.

5.1.2.3968 TIM4_AddressBase [35/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207R6.h.

5.1.2.3969 TIM4_AddressBase [36/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF5289.h.

5.1.2.3970 TIM4_AddressBase [37/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF6289.h.

5.1.2.3971 TIM4_AddressBase [38/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S208R6.h.

5.1.2.3972 TIM4_AddressBase [39/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207MB.h.

5.1.2.3973 TIM4_AddressBase [40/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF5269.h.

5.1.2.3974 TIM4_AddressBase [41/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF52A6.h.

5.1.2.3975 TIM4_AddressBase [42/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207S8.h.

5.1.2.3976 TIM4_AddressBase [43/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF62A8.h.

5.1.2.3977 TIM4_AddressBase [44/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S007C8.h.

5.1.2.3978 TIM4_AddressBase [45/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF628A.h.

5.1.2.3979 TIM4_AddressBase [46/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207C6.h.

5.1.2.3980 TIM4_AddressBase [47/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S208R8.h.

5.1.2.3981 TIM4_AddressBase [48/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S208RB.h.

5.1.2.3982 TIM4_AddressBase [49/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF62A6.h.

5.1.2.3983 TIM4_AddressBase [50/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207M8.h.

5.1.2.3984 TIM4_AddressBase [51/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207K8.h.

5.1.2.3985 TIM4_AddressBase [52/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S208S8.h.

5.1.2.3986 TIM4_AddressBase [53/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF52AA.h.

5.1.2.3987 TIM4_AddressBase [54/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF6286.h.

5.1.2.3988 TIM4_AddressBase [55/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S208C6.h.

5.1.2.3989 TIM4_AddressBase [56/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF6388.h.

5.1.2.3990 TIM4_AddressBase [57/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S208SB.h.

5.1.2.3991 TIM4_AddressBase [58/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S208S6.h.

5.1.2.3992 TIM4_AddressBase [59/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF5268.h.

5.1.2.3993 TIM4_AddressBase [60/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8S207C8.h.

5.1.2.3994 TIM4_AddressBase [61/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF52A9.h.

5.1.2.3995 TIM4_AddressBase [62/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF5288.h.

5.1.2.3996 TIM4_AddressBase [63/63]

```
#define TIM4_AddressBase 0x5340
```

Definition at line 77 of file STM8AF6288.h.

5.1.2.3997 TIM5_AddressBase [1/7]

```
#define TIM5_AddressBase 0x5300
```

Definition at line 71 of file STM8AF6213A.h.

5.1.2.3998 TIM5_AddressBase [2/7]

```
#define TIM5_AddressBase 0x5300
```

Definition at line 71 of file STM8AF6223.h.

5.1.2.3999 TIM5_AddressBase [3/7]

```
#define TIM5_AddressBase 0x5300
```

Definition at line 71 of file STM8S903F3.h.

5.1.2.4000 TIM5_AddressBase [4/7]

```
#define TIM5_AddressBase 0x5300
```

Definition at line 71 of file STM8AF6223A.h.

5.1.2.4001 TIM5_AddressBase [5/7]

```
#define TIM5_AddressBase 0x5300
```

Definition at line 71 of file STM8S903K3.h.

5.1.2.4002 TIM5_AddressBase [6/7]

```
#define TIM5_AddressBase 0x5300
```

Definition at line 71 of file STM8AF6226.h.

5.1.2.4003 TIM5_AddressBase [7/7]

```
#define TIM5_AddressBase 0x5300
```

Definition at line 71 of file STM8AF6213.h.

5.1.2.4004 TIM6_AddressBase [1/7]

```
#define TIM6_AddressBase 0x5340
```

Definition at line 72 of file STM8S903K3.h.

5.1.2.4005 TIM6_AddressBase [2/7]

```
#define TIM6_AddressBase 0x5340
```

Definition at line 72 of file STM8AF6213.h.

5.1.2.4006 TIM6_AddressBase [3/7]

```
#define TIM6_AddressBase 0x5340
```

Definition at line 72 of file STM8AF6226.h.

5.1.2.4007 TIM6_AddressBase [4/7]

```
#define TIM6_AddressBase 0x5340
```

Definition at line 72 of file STM8AF6213A.h.

5.1.2.4008 TIM6_AddressBase [5/7]

```
#define TIM6_AddressBase 0x5340
```

Definition at line 72 of file STM8S903F3.h.

5.1.2.4009 TIM6_AddressBase [6/7]

```
#define TIM6_AddressBase 0x5340
```

Definition at line 72 of file STM8AF6223A.h.

5.1.2.4010 TIM6_AddressBase [7/7]

```
#define TIM6_AddressBase 0x5340
```

Definition at line 72 of file STM8AF6223.h.

5.1.2.4011 TRIGGER_TRAP

```
#define TRIGGER_TRAP __asm__("trap")
```

trigger a trap (=soft interrupt) e.g. for EMC robustness (see AN1015)

Definition at line 171 of file STM8AF_STM8S.h.

5.1.2.4012 UART1_AddressBase [1/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 69 of file STM8S903K3.h.

5.1.2.4013 UART1_AddressBase [2/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 69 of file STM8S103F3.h.

5.1.2.4014 UART1_AddressBase [3/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 69 of file STM8S103F2.h.

5.1.2.4015 UART1_AddressBase [4/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 69 of file STM8S903F3.h.

5.1.2.4016 UART1_AddressBase [5/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 69 of file STM8S003K3.h.

5.1.2.4017 UART1_AddressBase [6/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 69 of file STM8S001J3.h.

5.1.2.4018 UART1_AddressBase [7/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 69 of file STM8S003F3.h.

5.1.2.4019 UART1_AddressBase [8/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 69 of file STM8S103K3.h.

5.1.2.4020 UART1_AddressBase [9/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF52A9.h.

5.1.2.4021 UART1_AddressBase [10/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF5269.h.

5.1.2.4022 UART1_AddressBase [11/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207C8.h.

5.1.2.4023 UART1_AddressBase [12/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207C6.h.

5.1.2.4024 UART1_AddressBase [13/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF62A9.h.

5.1.2.4025 UART1_AddressBase [14/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S208R6.h.

5.1.2.4026 UART1_AddressBase [15/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S208SB.h.

5.1.2.4027 UART1_AddressBase [16/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF62A8.h.

5.1.2.4028 UART1_AddressBase [17/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S208R8.h.

5.1.2.4029 UART1_AddressBase [18/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207MB.h.

5.1.2.4030 UART1_AddressBase [19/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S208S6.h.

5.1.2.4031 UART1_AddressBase [20/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF5268.h.

5.1.2.4032 UART1_AddressBase [21/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF6289.h.

5.1.2.4033 UART1_AddressBase [22/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S208CB.h.

5.1.2.4034 UART1_AddressBase [23/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF5289.h.

5.1.2.4035 UART1_AddressBase [24/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207S6.h.

5.1.2.4036 UART1_AddressBase [25/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207S8.h.

5.1.2.4037 UART1_AddressBase [26/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S208C6.h.

5.1.2.4038 UART1_AddressBase [27/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207CB.h.

5.1.2.4039 UART1_AddressBase [28/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF52A6.h.

5.1.2.4040 UART1_AddressBase [29/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF6286.h.

5.1.2.4041 UART1_AddressBase [30/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S208MB.h.

5.1.2.4042 UART1_AddressBase [31/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S208RB.h.

5.1.2.4043 UART1_AddressBase [32/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207SB.h.

5.1.2.4044 UART1_AddressBase [33/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207RB.h.

5.1.2.4045 UART1_AddressBase [34/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207K8.h.

5.1.2.4046 UART1_AddressBase [35/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF62AA.h.

5.1.2.4047 UART1_AddressBase [36/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF528A.h.

5.1.2.4048 UART1_AddressBase [37/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S208S8.h.

5.1.2.4049 UART1_AddressBase [38/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S007C8.h.

5.1.2.4050 UART1_AddressBase [39/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF52AA.h.

5.1.2.4051 UART1_AddressBase [40/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF62A6.h.

5.1.2.4052 UART1_AddressBase [41/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207R6.h.

5.1.2.4053 UART1_AddressBase [42/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S208C8.h.

5.1.2.4054 UART1_AddressBase [43/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF5286.h.

5.1.2.4055 UART1_AddressBase [44/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF52A8.h.

5.1.2.4056 UART1_AddressBase [45/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF628A.h.

5.1.2.4057 UART1_AddressBase [46/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF6388.h.

5.1.2.4058 UART1_AddressBase [47/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207R8.h.

5.1.2.4059 UART1_AddressBase [48/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207K6.h.

5.1.2.4060 UART1_AddressBase [49/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF5288.h.

5.1.2.4061 UART1_AddressBase [50/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8AF6288.h.

5.1.2.4062 UART1_AddressBase [51/51]

```
#define UART1_AddressBase 0x5230
```

Definition at line 72 of file STM8S207M8.h.

5.1.2.4063 UART2_AddressBase [1/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 69 of file STM8AF6366.h.

5.1.2.4064 UART2_AddressBase [2/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8S105K4.h.

5.1.2.4065 UART2_AddressBase [3/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8S105K6.h.

5.1.2.4066 UART2_AddressBase [4/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8AF6248.h.

5.1.2.4067 UART2_AddressBase [5/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8AF6268.h.

5.1.2.4068 UART2_AddressBase [6/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8S105C6.h.

5.1.2.4069 UART2_AddressBase [7/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8S005C6.h.

5.1.2.4070 UART2_AddressBase [8/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8S105C4.h.

5.1.2.4071 UART2_AddressBase [9/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8AF6266.h.

5.1.2.4072 UART2_AddressBase [10/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8S105S4.h.

5.1.2.4073 UART2_AddressBase [11/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8AF6269.h.

5.1.2.4074 UART2_AddressBase [12/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8S105S6.h.

5.1.2.4075 UART2_AddressBase [13/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8AF6246.h.

5.1.2.4076 UART2_AddressBase [14/14]

```
#define UART2_AddressBase 0x5240
```

Definition at line 70 of file STM8S005K6.h.

5.1.2.4077 UART3_AddressBase [1/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207CB.h.

5.1.2.4078 UART3_AddressBase [2/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207S8.h.

5.1.2.4079 UART3_AddressBase [3/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF52A8.h.

5.1.2.4080 UART3_AddressBase [4/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF6289.h.

5.1.2.4081 UART3_AddressBase [5/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S007C8.h.

5.1.2.4082 UART3_AddressBase [6/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF62A9.h.

5.1.2.4083 UART3_AddressBase [7/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF62A6.h.

5.1.2.4084 UART3_AddressBase [8/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF5286.h.

5.1.2.4085 UART3_AddressBase [9/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207MB.h.

5.1.2.4086 UART3_AddressBase [10/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S208S8.h.

5.1.2.4087 UART3_AddressBase [11/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF5288.h.

5.1.2.4088 UART3_AddressBase [12/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF6288.h.

5.1.2.4089 UART3_AddressBase [13/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF628A.h.

5.1.2.4090 UART3_AddressBase [14/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207K6.h.

5.1.2.4091 UART3_AddressBase [15/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF5289.h.

5.1.2.4092 UART3_AddressBase [16/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S208CB.h.

5.1.2.4093 UART3_AddressBase [17/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207R8.h.

5.1.2.4094 UART3_AddressBase [18/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF6286.h.

5.1.2.4095 UART3_AddressBase [19/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S208R8.h.

5.1.2.4096 UART3_AddressBase [20/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF5268.h.

5.1.2.4097 UART3_AddressBase [21/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S208C6.h.

5.1.2.4098 UART3_AddressBase [22/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF52A9.h.

5.1.2.4099 UART3_AddressBase [23/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S208SB.h.

5.1.2.4100 UART3_AddressBase [24/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF5269.h.

5.1.2.4101 UART3_AddressBase [25/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S208R6.h.

5.1.2.4102 UART3_AddressBase [26/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S208S6.h.

5.1.2.4103 UART3_AddressBase [27/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF62A8.h.

5.1.2.4104 UART3_AddressBase [28/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S208MB.h.

5.1.2.4105 UART3_AddressBase [29/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207C8.h.

5.1.2.4106 UART3_AddressBase [30/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207M8.h.

5.1.2.4107 UART3_AddressBase [31/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207RB.h.

5.1.2.4108 UART3_AddressBase [32/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207S6.h.

5.1.2.4109 UART3_AddressBase [33/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S208C8.h.

5.1.2.4110 UART3_AddressBase [34/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF52A6.h.

5.1.2.4111 UART3_AddressBase [35/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207C6.h.

5.1.2.4112 UART3_AddressBase [36/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207R6.h.

5.1.2.4113 UART3_AddressBase [37/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S208RB.h.

5.1.2.4114 UART3_AddressBase [38/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207K8.h.

5.1.2.4115 UART3_AddressBase [39/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF6388.h.

5.1.2.4116 UART3_AddressBase [40/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF528A.h.

5.1.2.4117 UART3_AddressBase [41/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF62AA.h.

5.1.2.4118 UART3_AddressBase [42/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8AF52AA.h.

5.1.2.4119 UART3_AddressBase [43/43]

```
#define UART3_AddressBase 0x5240
```

Definition at line 73 of file STM8S207SB.h.

5.1.2.4120 UART4_AddressBase [1/5]

```
#define UART4_AddressBase 0x5230
```

Definition at line 69 of file STM8AF6226.h.

5.1.2.4121 UART4_AddressBase [2/5]

```
#define UART4_AddressBase 0x5230
```

Definition at line 69 of file STM8AF6213.h.

5.1.2.4122 UART4_AddressBase [3/5]

```
#define UART4_AddressBase 0x5230
```

Definition at line 69 of file STM8AF6223.h.

5.1.2.4123 UART4_AddressBase [4/5]

```
#define UART4_AddressBase 0x5230
```

Definition at line 69 of file STM8AF6213A.h.

5.1.2.4124 UART4_AddressBase [5/5]

```
#define UART4_AddressBase 0x5230
```

Definition at line 69 of file STM8AF6223A.h.

5.1.2.4125 UID_AddressBase [1/34]

```
#define UID_AddressBase 0x4865
```

Definition at line 77 of file STM8S903F3.h.

5.1.2.4126 UID_AddressBase [2/34]

```
#define UID_AddressBase 0x4865
```

Definition at line 77 of file STM8S103K3.h.

5.1.2.4127 UID_AddressBase [3/34]

```
#define UID_AddressBase 0x4865
```

Definition at line 77 of file STM8S103F2.h.

5.1.2.4128 UID_AddressBase [4/34]

```
#define UID_AddressBase 0x4865
```

Definition at line 77 of file STM8S103F3.h.

5.1.2.4129 UID_AddressBase [5/34]

```
#define UID_AddressBase 0x4865
```

Definition at line 77 of file STM8S903K3.h.

5.1.2.4130 UID_AddressBase [6/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 79 of file STM8S105S4.h.

5.1.2.4131 UID_AddressBase [7/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 79 of file STM8S105C6.h.

5.1.2.4132 UID_AddressBase [8/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 79 of file STM8S105K6.h.

5.1.2.4133 UID_AddressBase [9/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 79 of file STM8S105K4.h.

5.1.2.4134 UID_AddressBase [10/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 79 of file STM8S105S6.h.

5.1.2.4135 UID_AddressBase [11/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 79 of file STM8S105C4.h.

5.1.2.4136 UID_AddressBase [12/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207R8.h.

5.1.2.4137 UID_AddressBase [13/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207R6.h.

5.1.2.4138 UID_AddressBase [14/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207C6.h.

5.1.2.4139 UID_AddressBase [15/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207M8.h.

5.1.2.4140 UID_AddressBase [16/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207SB.h.

5.1.2.4141 UID_AddressBase [17/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207S8.h.

5.1.2.4142 UID_AddressBase [18/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207K8.h.

5.1.2.4143 UID_AddressBase [19/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207RB.h.

5.1.2.4144 UID_AddressBase [20/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207CB.h.

5.1.2.4145 UID_AddressBase [21/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207K6.h.

5.1.2.4146 UID_AddressBase [22/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207C8.h.

5.1.2.4147 UID_AddressBase [23/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207MB.h.

5.1.2.4148 UID_AddressBase [24/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 82 of file STM8S207S6.h.

5.1.2.4149 UID_AddressBase [25/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 83 of file STM8S208MB.h.

5.1.2.4150 UID_AddressBase [26/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 83 of file STM8S208CB.h.

5.1.2.4151 UID_AddressBase [27/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 83 of file STM8S208C6.h.

5.1.2.4152 UID_AddressBase [28/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 83 of file STM8S208C8.h.

5.1.2.4153 UID_AddressBase [29/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 83 of file STM8S208S8.h.

5.1.2.4154 UID_AddressBase [30/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 83 of file STM8S208RB.h.

5.1.2.4155 UID_AddressBase [31/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 83 of file STM8S208R6.h.

5.1.2.4156 UID_AddressBase [32/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 83 of file STM8S208SB.h.

5.1.2.4157 UID_AddressBase [33/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 83 of file STM8S208S6.h.

5.1.2.4158 UID_AddressBase [34/34]

```
#define UID_AddressBase 0x48CD
```

Definition at line 83 of file STM8S208R8.h.

5.1.2.4159 WAIT_FOR_INTERRUPT

```
#define WAIT_FOR_INTERRUPT( ) __asm__("wfi")
```

stop code execution and wait for interrupt

Definition at line 172 of file STM8AF_STM8S.h.

5.1.2.4160 WWDG_AddressBase [1/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8AF6213A.h.

5.1.2.4161 WWDG_AddressBase [2/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8AF6226.h.

5.1.2.4162 WWDG_AddressBase [3/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8AF6223A.h.

5.1.2.4163 WWDG_AddressBase [4/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8S003K3.h.

5.1.2.4164 WWDG_AddressBase [5/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8S001J3.h.

5.1.2.4165 WWDG_AddressBase [6/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8AF6366.h.

5.1.2.4166 WWDG_AddressBase [7/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8S103F2.h.

5.1.2.4167 WWDG_AddressBase [8/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8S903F3.h.

5.1.2.4168 WWDG_AddressBase [9/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8AF6223.h.

5.1.2.4169 WWDG_AddressBase [10/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8S903K3.h.

5.1.2.4170 WWDG_AddressBase [11/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8AF6213.h.

5.1.2.4171 WWDG_AddressBase [12/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8S103F3.h.

5.1.2.4172 WWDG_AddressBase [13/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8S003F3.h.

5.1.2.4173 WWDG_AddressBase [14/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 63 of file STM8S103K3.h.

5.1.2.4174 WWDG_AddressBase [15/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8S005C6.h.

5.1.2.4175 WWDG_AddressBase [16/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8S105C6.h.

5.1.2.4176 WWDG_AddressBase [17/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8AF6246.h.

5.1.2.4177 WWDG_AddressBase [18/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8S005K6.h.

5.1.2.4178 WWDG_AddressBase [19/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8S105S6.h.

5.1.2.4179 WWDG_AddressBase [20/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8AF6269.h.

5.1.2.4180 WWDG_AddressBase [21/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8S105K6.h.

5.1.2.4181 WWDG_AddressBase [22/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8S105S4.h.

5.1.2.4182 WWDG_AddressBase [23/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8S105C4.h.

5.1.2.4183 WWDG_AddressBase [24/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8AF6268.h.

5.1.2.4184 WWDG_AddressBase [25/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8AF6248.h.

5.1.2.4185 WWDG_AddressBase [26/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8S105K4.h.

5.1.2.4186 WWDG_AddressBase [27/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 64 of file STM8AF6266.h.

5.1.2.4187 WWDG_AddressBase [28/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207C6.h.

5.1.2.4188 WWDG_AddressBase [29/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF5268.h.

5.1.2.4189 WWDG_AddressBase [30/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207S8.h.

5.1.2.4190 WWDG_AddressBase [31/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S208R6.h.

5.1.2.4191 WWDG_AddressBase [32/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF6288.h.

5.1.2.4192 WWDG_AddressBase [33/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207S6.h.

5.1.2.4193 WWDG_AddressBase [34/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207MB.h.

5.1.2.4194 WWDG_AddressBase [35/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207SB.h.

5.1.2.4195 WWDG_AddressBase [36/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S208C8.h.

5.1.2.4196 WWDG_AddressBase [37/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF62A8.h.

5.1.2.4197 WWDG_AddressBase [38/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207CB.h.

5.1.2.4198 WWDG_AddressBase [39/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF5286.h.

5.1.2.4199 WWDG_AddressBase [40/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF62AA.h.

5.1.2.4200 WWDG_AddressBase [41/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S007C8.h.

5.1.2.4201 WWDG_AddressBase [42/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF528A.h.

5.1.2.4202 WWDG_AddressBase [43/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF52A8.h.

5.1.2.4203 WWDG_AddressBase [44/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207R6.h.

5.1.2.4204 WWDG_AddressBase [45/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S208CB.h.

5.1.2.4205 WWDG_AddressBase [46/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF6388.h.

5.1.2.4206 WWDG_AddressBase [47/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207M8.h.

5.1.2.4207 WWDG_AddressBase [48/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF6289.h.

5.1.2.4208 WWDG_AddressBase [49/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S208MB.h.

5.1.2.4209 WWDG_AddressBase [50/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF62A9.h.

5.1.2.4210 WWDG_AddressBase [51/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207RB.h.

5.1.2.4211 WWDG_AddressBase [52/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S208R8.h.

5.1.2.4212 WWDG_AddressBase [53/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF5288.h.

5.1.2.4213 WWDG_AddressBase [54/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S208SB.h.

5.1.2.4214 WWDG_AddressBase [55/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF628A.h.

5.1.2.4215 WWDG_AddressBase [56/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207K6.h.

5.1.2.4216 WWDG_AddressBase [57/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF52A6.h.

5.1.2.4217 WWDG_AddressBase [58/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF5289.h.

5.1.2.4218 WWDG_AddressBase [59/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S208S6.h.

5.1.2.4219 WWDG_AddressBase [60/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF5269.h.

5.1.2.4220 WWDG_AddressBase [61/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207K8.h.

5.1.2.4221 WWDG_AddressBase [62/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207R8.h.

5.1.2.4222 WWDG_AddressBase [63/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF6286.h.

5.1.2.4223 WWDG_AddressBase [64/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF52AA.h.

5.1.2.4224 WWDG_AddressBase [65/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S207C8.h.

5.1.2.4225 WWDG_AddressBase [66/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S208S8.h.

5.1.2.4226 WWDG_AddressBase [67/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF62A6.h.

5.1.2.4227 WWDG_AddressBase [68/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S208C6.h.

5.1.2.4228 WWDG_AddressBase [69/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8S208RB.h.

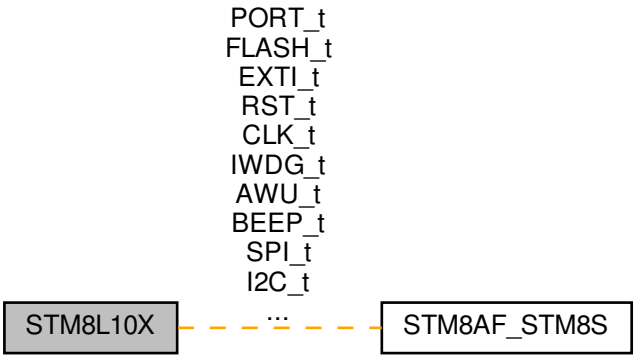
5.1.2.4229 WWDG_AddressBase [70/70]

```
#define WWDG_AddressBase 0x50D1
```

Definition at line 66 of file STM8AF52A9.h.

5.2 STM8L10X

Collaboration diagram for STM8L10X:



Data Structures

- struct PORT_t

  structure for controlling pins in PORT mode (PORTx, x=A..I)
- struct FLASH_t

  struct to control write/erase of flash memory (FLASH)
- struct EXTI_t

  struct for configuring external port interrupts (EXTI)
- struct RST_t

  struct for determining reset source (RST)
- struct CLK_t

  struct for configuring/monitoring clock module (CLK)
- struct IWDG_t

  struct for access to Independent Timeout Watchdog registers (IWDG)
- struct AWU_t

  struct for cofiguring the Auto Wake-Up Module (AWU)
- struct BEEP_t

  struct for beeper control (BEEP)
- struct SPI_t

  struct for controlling SPI module (SPI)
- struct I2C_t

  struct for controlling I2C module (I2C)
- struct USART_t

  struct for controlling Universal Asynchronous Receiver Transmitter (USART)
- struct WFE_t

  struct to configure interrupt sources as external interrupts or wake events (WFE)
- struct TIM2_3_t

  struct for controlling 16-Bit Timer 2+3 (TIM2, TIM3)

- struct TIM4_t

  struct for controlling 8-Bit Timer 4 (TIM4)
- struct IRTIM_t

  struct for Infrared Timer Module (IRTIM)
- struct COMP_t

  struct for Comparator Module (COMP)
- struct CFG_t

  struct for Global Configuration registers (CFG)
- struct ITC_t

  struct for setting interrupt Priority (ITC)

Macros

- #define STM8L101F1
- #define STM8L10x
- #define STM8_PFLASH_SIZE 2048
- #define STM8_RAM_SIZE 1536
- #define STM8_EEPROM_SIZE 0
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define FLASH_AddressBase 0x5050
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B0
- #define CLK_AddressBase 0x50C0
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define USART_AddressBase 0x5230
- #define WFE_AddressBase 0x50A6
- #define TIM2_AddressBase 0x5250
- #define TIM3_AddressBase 0x5280
- #define TIM4_AddressBase 0x52E0
- #define IRTIM_AddressBase 0x52FF
- #define COMP_AddressBase 0x5300
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4925
- #define STM8L101F2
- #define STM8L10x
- #define STM8_PFLASH_SIZE 4096
- #define STM8_RAM_SIZE 1536
- #define STM8_EEPROM_SIZE 0
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F

- #define FLASH_AddressBase 0x5050
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B0
- #define CLK_AddressBase 0x50C0
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define USART_AddressBase 0x5230
- #define WFE_AddressBase 0x50A6
- #define TIM2_AddressBase 0x5250
- #define TIM3_AddressBase 0x5280
- #define TIM4_AddressBase 0x52E0
- #define IRTIM_AddressBase 0x52FF
- #define COMP_AddressBase 0x5300
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4925
- #define STM8L101F3
- #define STM8L10x
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1536
- #define STM8_EEPROM_SIZE 0
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define FLASH_AddressBase 0x5050
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B0
- #define CLK_AddressBase 0x50C0
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define USART_AddressBase 0x5230
- #define WFE_AddressBase 0x50A6
- #define TIM2_AddressBase 0x5250
- #define TIM3_AddressBase 0x5280
- #define TIM4_AddressBase 0x52E0
- #define IRTIM_AddressBase 0x52FF
- #define COMP_AddressBase 0x5300
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4925
- #define STM8L101G2
- #define STM8L10x
- #define STM8_PFLASH_SIZE 4096
- #define STM8_RAM_SIZE 1536
- #define STM8_EEPROM_SIZE 0

- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define FLASH_AddressBase 0x5050
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B0
- #define CLK_AddressBase 0x50C0
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define USART_AddressBase 0x5230
- #define WFE_AddressBase 0x50A6
- #define TIM2_AddressBase 0x5250
- #define TIM3_AddressBase 0x5280
- #define TIM4_AddressBase 0x52E0
- #define IRTIM_AddressBase 0x52FF
- #define COMP_AddressBase 0x5300
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4925
- #define STM8L101G3
- #define STM8L10x
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1536
- #define STM8_EEPROM_SIZE 0
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define FLASH_AddressBase 0x5050
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B0
- #define CLK_AddressBase 0x50C0
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define USART_AddressBase 0x5230
- #define WFE_AddressBase 0x50A6
- #define TIM2_AddressBase 0x5250
- #define TIM3_AddressBase 0x5280
- #define TIM4_AddressBase 0x52E0
- #define IRTIM_AddressBase 0x52FF
- #define COMP_AddressBase 0x5300
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4925

- #define STM8L101K3
- #define STM8L10x
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1536
- #define STM8_EEPROM_SIZE 0
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define FLASH_AddressBase 0x5050
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B0
- #define CLK_AddressBase 0x50C0
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define USART_AddressBase 0x5230
- #define WFE_AddressBase 0x50A6
- #define TIM2_AddressBase 0x5250
- #define TIM3_AddressBase 0x5280
- #define TIM4_AddressBase 0x52E0
- #define IRTIM_AddressBase 0x52FF
- #define COMP_AddressBase 0x5300
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4925
- #define STM8_PFLASH_SIZE 2048

  size of program flash [B]
- #define STM8_RAM_SIZE 1536

  size of RAM [B]
- #define STM8_EEPROM_SIZE 0

  size of data EEPROM [B]
- #define STM8_PFLASH_START 0x8000

  first address in program flash
- #define STM8_PFLASH_END (STM8_PFLASH_START + STM8_PFLASH_SIZE - 1)

  last address in program flash
- #define STM8_RAM_START 0x0000

  first address in RAM
- #define STM8_RAM_END (STM8_RAM_START + STM8_RAM_SIZE - 1)

  last address in RAM
- #define STM8_EEPROM_START 9800

  first address in EEPROM
- #define STM8_EEPROM_END (STM8_EEPROM_START + STM8_EEPROM_SIZE - 1)

  last address in EEPROM
- #define STM8_ADDR_WIDTH 16

  width of address space
- #define STM8_MEM_POINTER_T uint16_t

  address variable type
- #define ISR_HANDLER(func, irq) void func(void) __interrupt(irq)

handler for interrupt service routine

- #define ISR_HANDLER_TRAP(func) void func() __trap

  handler for trap service routine
- #define NOP() __asm__("nop")

  perform a nop() operation (=minimum delay)
- #define DISABLE_INTERRUPTS() __asm__("sim")

  disable interrupt handling
- #define ENABLE_INTERRUPTS() __asm__("rim")

  enable interrupt handling
- #define TRIGGER_TRAP __asm__("trap")

  trigger a trap (=soft interrupt) e.g. for EMC robustness (see AN1015)
- #define WAIT_FOR_INTERRUPT() __asm__("wfi")

  stop code execution and wait for interrupt
- #define ENTER_HALT() __asm__("halt")

  put controller to HALT mode
- #define SW_RESET() (_WWDG_CR=0xBF)

  reset controller via WWGD module
- #define _BITS unsigned int

  data type in bit structs (follow C90 standard)
- #define _SFR(type, addr) (*((volatile type*) (addr)))

  peripheral register
- #define __FLASH_VECTOR__ 1

  irq1 - flash interrupt
- #define __AWU_VECTOR__ 4

  irq4 - Auto Wake Up from Halt interrupt (AWU)
- #define __PORTB_VECTOR__ 6

  irq6 - External interrupt port B
- #define __PORTD_VECTOR__ 7

  irq7 - External interrupt port D
- #define __EXTI0_VECTOR__ 8

  irq8 - External interrupt 0
- #define __EXTI1_VECTOR__ 9

  irq9 - External interrupt 1
- #define __EXTI2_VECTOR__ 10

  irq10 - External interrupt 2
- #define __EXTI3_VECTOR__ 11

  irq11 - External interrupt 3
- #define __EXTI4_VECTOR__ 12

  irq12 - External interrupt 4
- #define __EXTI5_VECTOR__ 13

  irq13 - External interrupt 5
- #define __EXTI6_VECTOR__ 14

  irq14 - External interrupt 6
- #define __EXTI7_VECTOR__ 15

  irq15 - External interrupt 7
- #define __COMP_VECTOR__ 18

  irq18 - comparator interrupt
- #define __TIM2_UPD_OVF_VECTOR__ 19

  irq19 - TIM2 Update/overflow/trigger/break interrupt
- #define __TIM2_CAPCOM_VECTOR__ 20

  irq20 - TIM2 Capture/Compare interrupt

- #define __TIM3_UPD_OVF_VECTOR__ 21
  irq21 - TIM3 Update/overflow/break interrupt
- #define __TIM3_CAPCOM_VECTOR__ 22
  irq22 - TIM3 Capture/Compare interrupt
- #define __TIM4_UPD_VECTOR__ 25
  irq25 - TIM4 Update/trigger interrupt
- #define __SPI_VECTOR__ 26
  irq26 - SPI End of transfer interrupt
- #define __USART_TXE_VECTOR__ 27
  irq27 - USART send (TX empty) interrupt
- #define __USART_RXF_VECTOR__ 28
  irq28 - USART receive (RX full) interrupt
- #define __I2C_VECTOR__ 19
  irq29 - I2C interrupt
- #define _GPIOA _SFR(PORT_t, PORTA_AddressBase)
  port A struct/bit access
- #define _GPIOA_ODR _SFR(uint8_t, PORTA_AddressBase+0x00)
  port A output register
- #define _GPIOA_IDR _SFR(uint8_t, PORTA_AddressBase+0x01)
  port A input register
- #define _GPIOA_DDR _SFR(uint8_t, PORTA_AddressBase+0x02)
  port A direction register
- #define _GPIOA_CR1 _SFR(uint8_t, PORTA_AddressBase+0x03)
  port A control register 1
- #define _GPIOA_CR2 _SFR(uint8_t, PORTA_AddressBase+0x04)
  port A control register 2
- #define _GPIOB _SFR(PORT_t, PORTB_AddressBase)
  port B struct/bit access
- #define _GPIOB_ODR _SFR(uint8_t, PORTB_AddressBase+0x00)
  port B output register
- #define _GPIOB_IDR _SFR(uint8_t, PORTB_AddressBase+0x01)
  port B input register
- #define _GPIOB_DDR _SFR(uint8_t, PORTB_AddressBase+0x02)
  port B direction register
- #define _GPIOB_CR1 _SFR(uint8_t, PORTB_AddressBase+0x03)
  port B control register 1
- #define _GPIOB_CR2 _SFR(uint8_t, PORTB_AddressBase+0x04)
  port B control register 2
- #define _GPIOC _SFR(PORT_t, PORTC_AddressBase)
  port C struct/bit access
- #define _GPIOC_ODR _SFR(uint8_t, PORTC_AddressBase+0x00)
  port C output register
- #define _GPIOC_IDR _SFR(uint8_t, PORTC_AddressBase+0x01)
  port C input register
- #define _GPIOC_DDR _SFR(uint8_t, PORTC_AddressBase+0x02)
  port C direction register
- #define _GPIOC_CR1 _SFR(uint8_t, PORTC_AddressBase+0x03)
  port C control register 1
- #define _GPIOC_CR2 _SFR(uint8_t, PORTC_AddressBase+0x04)
  port C control register 2
- #define _GPIOD _SFR(PORT_t, PORTD_AddressBase)

port D struct/bit access

- #define _GPIOD_ODR _SFR(uint8_t, PORTD_AddressBase+0x00)

  port D output register
- #define _GPIOD_IDR _SFR(uint8_t, PORTD_AddressBase+0x01)

  port D input register
- #define _GPIOD_DDR _SFR(uint8_t, PORTD_AddressBase+0x02)

  port D direction register
- #define _GPIOD_CR1 _SFR(uint8_t, PORTD_AddressBase+0x03)

  port D control register 1
- #define _GPIOD_CR2 _SFR(uint8_t, PORTD_AddressBase+0x04)

  port D control register 2
- #define _GPIO_ODR_RESET_VALUE ((uint8_t) 0x00)

  port output register reset value
- #define _GPIO_DDR_RESET_VALUE ((uint8_t) 0x00)

  port direction register reset value
- #define _GPIO_CR1_RESET_VALUE ((uint8_t) 0x00)

  port control register 1 reset value
- #define _GPIO_CR2_RESET_VALUE ((uint8_t) 0x00)

  port control register 2 reset value
- #define _GPIO_PIN0 ((uint8_t) (0x01 << 0))

  port bit mask for pin 0 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN1 ((uint8_t) (0x01 << 1))

  port bit mask for pin 1 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN2 ((uint8_t) (0x01 << 2))

  port bit mask for pin 2 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN3 ((uint8_t) (0x01 << 3))

  port bit mask for pin 3 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN4 ((uint8_t) (0x01 << 4))

  port bit mask for pin 4 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN5 ((uint8_t) (0x01 << 5))

  port bit mask for pin 5 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN6 ((uint8_t) (0x01 << 6))

  port bit mask for pin 6 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN7 ((uint8_t) (0x01 << 7))

  port bit mask for pin 7 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _FLASH _SFR(FLASH_t, FLASH_AddressBase)

  Flash struct/bit access.
- #define _FLASH_CR1 _SFR(uint8_t, FLASH_AddressBase+0x00)

  Flash control register 1 (FLASH_CR1)
- #define _FLASH_CR2 _SFR(uint8_t, FLASH_AddressBase+0x01)

  Flash control register 2 (FLASH_CR2)
- #define _FLASH_PUKR _SFR(uint8_t, FLASH_AddressBase+0x02)

  Flash program memory unprotecting key register (FLASH_PUKR)
- #define _FLASH_DUKR _SFR(uint8_t, FLASH_AddressBase+0x03)

  Data EEPROM unprotection key register (FLASH_DUKR)
- #define _FLASH_IAPSR _SFR(uint8_t, FLASH_AddressBase+0x04)

  Flash status register (FLASH_IAPSR)
- #define _FLASH_CR1_RESET_VALUE ((uint8_t) 0x00)

  Flash control register 1 reset value.
- #define _FLASH_CR2_RESET_VALUE ((uint8_t) 0x00)

  Flash control register 2 reset value.

- #define _FLASH_PUKR_RESET_VALUE ((uint8_t) 0x00)

  Flash program memory unprotecting key reset value.
- #define _FLASH_DUKR_RESET_VALUE ((uint8_t) 0x00)

  Data EEPROM unprotection key reset value.
- #define _FLASH_IAPSR_RESET_VALUE ((uint8_t) 0x40)

  Flash status register reset value.
- #define _FLASH_FIX ((uint8_t) (0x01 << 0))

  Fixed Byte programming time [0] (in _FLASH_CR1)
- #define _FLASH_IE ((uint8_t) (0x01 << 1))

  Flash Interrupt enable [0] (in _FLASH_CR1)
- #define _FLASH_PRG ((uint8_t) (0x01 << 0))

  Standard block programming [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_FPRG ((uint8_t) (0x01 << 4))

  Fast block programming [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_ERASE ((uint8_t) (0x01 << 5))

  Block erasing [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_WPRG ((uint8_t) (0x01 << 6))

  Word programming [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_OPT ((uint8_t) (0x01 << 7))

  Write option bytes [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_WR_PG_DIS ((uint8_t) (0x01 << 0))

  Write attempted to protected page flag [0] (in _FLASH_IAPSR)
- #define _FLASH_PUL ((uint8_t) (0x01 << 1))

  Flash Program memory unlocked flag [0] (in _FLASH_IAPSR)
- #define _FLASH_EOP ((uint8_t) (0x01 << 2))

  End of programming (write or erase operation) flag [0] (in _FLASH_IAPSR)
- #define _FLASH_DUL ((uint8_t) (0x01 << 3))

  Data EEPROM area unlocked flag [0] (in _FLASH_IAPSR)
- #define _EXTI _SFR(EXTI_t, EXTI_AddressBase)

  External interrupt struct/bit access.
- #define _EXTI_CR1 _SFR(uint8_t, EXTI_AddressBase+0x00)

  External interrupt control register 1 (EXTI_CR1)
- #define _EXTI_CR2 _SFR(uint8_t, EXTI_AddressBase+0x01)

  External interrupt control register 2 (EXTI_CR2)
- #define _EXTI_CR3 _SFR(uint8_t, EXTI_AddressBase+0x02)

  External interrupt control register 3 (EXTI_CR2)
- #define _EXTI_SR1 _SFR(uint8_t, EXTI_AddressBase+0x03)

  External interrupt status register 1 (EXTI_SR1)
- #define _EXTI_SR2 _SFR(uint8_t, EXTI_AddressBase+0x04)

  External interrupt status register 2 (EXTI_SR2)
- #define _EXTI_CONF _SFR(uint8_t, EXTI_AddressBase+0x05)

  External interrupt port selector (EXTI_CONF)
- #define _EXTI_CR1_RESET_VALUE ((uint8_t) 0x00)

  External interrupt control register 1 reset value.
- #define _EXTI_CR2_RESET_VALUE ((uint8_t) 0x00)

  External interrupt control register 2 reset value.
- #define _EXTI_CR3_RESET_VALUE ((uint8_t) 0x00)

  External interrupt control register 3 reset value.
- #define _EXTI_SR1_RESET_VALUE ((uint8_t) 0x00)

  External interrupt status register 1 reset value.
- #define _EXTI_SR2_RESET_VALUE ((uint8_t) 0x00)

External interrupt status register 2 reset value.

- #define _EXTI_CONF_RESET_VALUE ((uint8_t) 0x00)

  External interrupt port selector reset value.
- #define _EXTI_P0IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 0 [1:0] (in _EXTI_CR1)
- #define _EXTI_P0IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 0 [0] (in _EXTI_CR1)
- #define _EXTI_P0IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 0 [1] (in _EXTI_CR1)
- #define _EXTI_P1IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 1 [1:0] (in _EXTI_CR1)
- #define _EXTI_P1IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 1 [0] (in _EXTI_CR1)
- #define _EXTI_P1IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 1 [1] (in _EXTI_CR1)
- #define _EXTI_P2IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 2 [1:0] (in _EXTI_CR1)
- #define _EXTI_P2IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 2 [0] (in _EXTI_CR1)
- #define _EXTI_P2IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 2 [1] (in _EXTI_CR1)
- #define _EXTI_P3IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 3 [1:0] (in _EXTI_CR1)
- #define _EXTI_P3IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 3 [0] (in _EXTI_CR1)
- #define _EXTI_P3IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 3 [1] (in _EXTI_CR1)
- #define _EXTI_P4IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 4 [1:0] (in _EXTI_CR2)
- #define _EXTI_P4IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 4 [0] (in _EXTI_CR2)
- #define _EXTI_P4IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 4 [1] (in _EXTI_CR2)
- #define _EXTI_P5IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 5 [1:0] (in _EXTI_CR2)
- #define _EXTI_P5IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 5 [0] (in _EXTI_CR2)
- #define _EXTI_P5IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 5 [1] (in _EXTI_CR2)
- #define _EXTI_P6IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 6 [1:0] (in _EXTI_CR2)
- #define _EXTI_P6IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 6 [0] (in _EXTI_CR2)
- #define _EXTI_P6IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 6 [1] (in _EXTI_CR2)
- #define _EXTI_P7IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 7 [1:0] (in _EXTI_CR2)
- #define _EXTI_P7IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 7 [0] (in _EXTI_CR2)
- #define _EXTI_P7IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 7 [1] (in _EXTI_CR2)

- #define _EXTI_PBIS ((uint8_t) (0x03 << 0))

  Port B external interrupt sensitivity bits [1:0] (in _EXTI_CR3)
- #define _EXTI_PBIS0 ((uint8_t) (0x01 << 0))

  Port B external interrupt sensitivity bits [0] (in _EXTI_CR3)
- #define _EXTI_PBIS1 ((uint8_t) (0x01 << 1))

  Port B external interrupt sensitivity bits [1] (in _EXTI_CR3)
- #define _EXTI_PDIS ((uint8_t) (0x03 << 0))

  Port D external interrupt sensitivity bits [1:0] (in _EXTI_CR3)
- #define _EXTI_PDIS0 ((uint8_t) (0x01 << 0))

  Port D external interrupt sensitivity bits [0] (in _EXTI_CR3)
- #define _EXTI_PDIS1 ((uint8_t) (0x01 << 1))

  Port D external interrupt sensitivity bits [1] (in _EXTI_CR3)
- #define _EXTI_P0F ((uint8_t) (0x01 << 0))

  Portx bit 0 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P1F ((uint8_t) (0x02 << 0))

  Portx bit 1 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P2F ((uint8_t) (0x04 << 0))

  Portx bit 2 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P3F ((uint8_t) (0x08 << 0))

  Portx bit 3 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P4F ((uint8_t) (0x10 << 0))

  Portx bit 4 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P5F ((uint8_t) (0x20 << 0))

  Portx bit 5 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P6F ((uint8_t) (0x40 << 0))

  Portx bit 6 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P7F ((uint8_t) (0x80 << 0))

  Portx bit 7 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_PBF ((uint8_t) (0x01 << 0))

  Port B external interrupt flag (in _EXTI_SR2)
- #define _EXTI_PDF ((uint8_t) (0x02 << 0))

  Port D external interrupt flag (in _EXTI_SR2)
- #define _EXTI_PBLIS ((uint8_t) (0x01 << 0))

  Port B, pins 0..3 external interrupt select (in _EXTI_CONF)
- #define _EXTI_PBHIS ((uint8_t) (0x02 << 0))

  Port B, pins 4..7 external interrupt select (in _EXTI_CONF)
- #define _EXTI_PDLIS ((uint8_t) (0x01 << 0))

  Port D, pins 0..3 external interrupt select (in _EXTI_CONF)
- #define _EXTI_PDHIS ((uint8_t) (0x02 << 0))

  Port D, pins 4..7 external interrupt select (in _EXTI_CONF)
- #define _RST _SFR(RST_t, RST_AddressBase)

  Reset module struct/bit access.
- #define _RST_CR _SFR(uint8_t, RST_AddressBase+0x00)

  Reset pin configuration register (RST_CR)
- #define _RST_SR _SFR(uint8_t, RST_AddressBase+0x01)

  Reset module status register (RST_SR)
- #define _RST_PIN_KEY ((uint8_t) 0xD0)

  Configure PA1 as GPIO, else NRST (in _RST_CR)
- #define _RST_PORF ((uint8_t) (0x01 << 0))

  Power-on reset (POR) flag [0] (in _RST_SR)
- #define _RST_IWDGF ((uint8_t) (0x01 << 1))

Independent Watchdog reset flag [0] (in _RST_SR)

- #define _RST_ILLOPF ((uint8_t) (0x01 << 2))

  Illegal opcode reset flag [0] (in _RST_SR)
- #define _RST_SWIMF ((uint8_t) (0x01 << 3))

  SWIM reset flag [0] (in _RST_SR)
- #define _CLK _SFR(CLK_t, CLK_AddressBase)

  Clock module struct/bit access.
- #define _CLK_CKDIVR _SFR(uint8_t, CLK_AddressBase+0x00)

  Clock Divider Register.
- #define _CLK_PCKENR _SFR(uint8_t, CLK_AddressBase+0x03)

  Peripheral clock gating register.
- #define _CLK_CCOR _SFR(uint8_t, CLK_AddressBase+0x05)

  Configurable clock output register.
- #define _CLK_CKDIVR_RESET_VALUE ((uint8_t) 0x03)

  Clock divider register reset value.
- #define _CLK_PCKENR_RESET_VALUE ((uint8_t) 0x00)

  Peripheral clock gating register reset value.
- #define _CLK_CCOR_RESET_VALUE ((uint8_t) 0x00)

  Configurable clock output register reset value.
- #define _CLK_HSIDIV ((uint8_t) (0x03 << 0))

  High speed internal clock prescaler [1:0] (in _CLK_CKDIVR)
- #define _CLK_HSIDIV0 ((uint8_t) (0x01 << 0))

  High speed internal clock prescaler [0] (in _CLK_CKDIVR)
- #define _CLK_HSIDIV1 ((uint8_t) (0x01 << 1))

  High speed internal clock prescaler [1] (in _CLK_CKDIVR)
- #define _CLK_HSIDIV_DIV1 ((uint8_t) (0x00 << 0))

  set HSI prescaler to 1 (in _CLK_CKDIVR)
- #define _CLK_HSIDIV_DIV2 ((uint8_t) (0x01 << 0))

  set HSI prescaler to 1/2 (in _CLK_CKDIVR)
- #define _CLK_HSIDIV_DIV4 ((uint8_t) (0x02 << 0))

  set HSI prescaler to 1/4 (in _CLK_CKDIVR)
- #define _CLK_HSIDIV_DIV8 ((uint8_t) (0x03 << 0))

  set HSI prescaler to 1/8 (in _CLK_CKDIVR)
- #define _CLK_TIM2 ((uint8_t) (0x01 << 0))

  clock enable TIM2 [0] (in _CLK_PCKENR1)
- #define _CLK_TIM3 ((uint8_t) (0x01 << 1))

  clock enable TIM3 [0] (in _CLK_PCKENR1)
- #define _CLK_TIM4 ((uint8_t) (0x01 << 2))

  clock enable TIM4 [0] (in _CLK_PCKENR1)
- #define _CLK_I2C ((uint8_t) (0x01 << 3))

  clock enable I2C [0] (in _CLK_PCKENR1)
- #define _CLK_SPI ((uint8_t) (0x01 << 4))

  clock enable SPI [0] (in _CLK_PCKENR1)
- #define _CLK_USART ((uint8_t) (0x01 << 5))

  clock enable USART [0] (in _CLK_PCKENR1)
- #define _CLK_AWU_BEEP ((uint8_t) (0x01 << 6))

  clock enable AWU/BEEP [0] (in _CLK_PCKENR1)
- #define _CLK_CCOEN ((uint8_t) (0x01 << 0))

  Configurable clock output enable [0] (in _CLK_CCOR)
- #define _CLK_CCOSEL ((uint8_t) (0x03 << 1))

  Configurable clock output selection [1:0] (in _CLK_CCOR)

- #define _CLK_CCOSEL0 ((uint8_t) (0x01 << 1))

  Configurable clock output selection [0] (in _CLK_CCOR)
- #define _CLK_CCOSEL1 ((uint8_t) (0x01 << 2))

  Configurable clock output selection [1] (in _CLK_CCOR)
- #define _CLK_CCOSEL_DIV1 ((uint8_t) (0x00 << 1))

  set clock output selection to 1 (in _CLK_CCOR)
- #define _CLK_CCOSEL_DIV2 ((uint8_t) (0x01 << 1))

  set clock output selection to 1/2 (in _CLK_CCOR)
- #define _CLK_CCOSEL_DIV4 ((uint8_t) (0x02 << 1))

  set clock output selection to 1/4 (in _CLK_CCOR)
- #define _CLK_CCOSEL_DIV16 ((uint8_t) (0x03 << 1))

  set clock output selection to 1/16 (in _CLK_CCOR)
- #define _IWDG _SFR(IWDG_t, IWDG_AddressBase)

  Independent Timeout Watchdog struct/bit access.
- #define _IWDG_KR _SFR(uint8_t, IWDG_AddressBase+0x00)

  Independent Timeout Watchdog Key register (IWDG_KR)
- #define _IWDG_PR _SFR(uint8_t, IWDG_AddressBase+0x01)

  Independent Timeout Watchdog Prescaler register (IWDG_PR)
- #define _IWDG_RLR _SFR(uint8_t, IWDG_AddressBase+0x02)

  Independent Timeout Watchdog Reload register (IWDG_RLR)
- #define _IWDG_PR_RESET_VALUE ((uint8_t) 0x00)

  Independent Timeout Watchdog Prescaler register reset value.
- #define _IWDG_RLR_RESET_VALUE ((uint8_t) 0xFF)

  Independent Timeout Watchdog Reload register reset value.
- #define _IWDG_KEY_ENABLE ((uint8_t) 0xCC)

  Independent Timeout Watchdog enable (in _IWDG_KR)
- #define _IWDG_KEY_REFRESH ((uint8_t) 0xAA)

  Independent Timeout Watchdog refresh (in _IWDG_KR)
- #define _IWDG_KEY_ACCESS ((uint8_t) 0x55)

  Independent Timeout Watchdog unlock write to IWDG_PR and IWDG_RLR (in _IWDG_KR)
- #define _IWDG_PRE ((uint8_t) (0x07 << 0))

  Independent Timeout Watchdog Prescaler divider [2:0] (in _IWDG_PR)
- #define _IWDG_PRE0 ((uint8_t) (0x01 << 0))

  Independent Timeout Watchdog Prescaler divider [0] (in _IWDG_PR)
- #define _IWDG_PRE1 ((uint8_t) (0x01 << 1))

  Independent Timeout Watchdog Prescaler divider [1] (in _IWDG_PR)
- #define _IWDG_PRE2 ((uint8_t) (0x01 << 2))

  Independent Timeout Watchdog Prescaler divider [2] (in _IWDG_PR)
- #define _AWU _SFR(AWU_t, AWU_AddressBase)

  Auto Wake-Up struct/bit access.
- #define _AWU_CSR _SFR(uint8_t, AWU_AddressBase+0x00)

  Auto Wake-Up Control/status register (AWU_CSR)
- #define _AWU_APR _SFR(uint8_t, AWU_AddressBase+0x01)

  Auto Wake-Up Asynchronous prescaler register (AWU_APR)
- #define _AWU_TBR _SFR(uint8_t, AWU_AddressBase+0x02)

  Auto Wake-Up Timebase selection register (AWU_TBR)
- #define _AWU_CSR_RESET_VALUE ((uint8_t) 0x00)

  Auto Wake-Up Control/status register reset value.
- #define _AWU_APR_RESET_VALUE ((uint8_t) 0x3F)

  Auto Wake-Up Asynchronous prescaler register reset value.
- #define _AWU_TBR_RESET_VALUE ((uint8_t) 0x00)

Auto Wake-Up Timebase selection register reset value.

- #define _AWU_MSR ((uint8_t) (0x01 << 0))

  Auto Wake-Up LSI measurement enable [0] (in _AWU_CSR)
- #define _AWU_AWUEN ((uint8_t) (0x01 << 4))

  Auto-wakeup enable [0] (in _AWU_CSR)
- #define _AWU_AWUF ((uint8_t) (0x01 << 5))

  Auto-wakeup status flag [0] (in _AWU_CSR)
- #define _AWU_APRE ((uint8_t) (0x3F << 0))

  Auto-wakeup asynchronous prescaler divider [5:0] (in _AWU_APR)
- #define _AWU_APRE0 ((uint8_t) (0x01 << 0))

  Auto-wakeup asynchronous prescaler divider [0] (in _AWU_APR)
- #define _AWU_APRE1 ((uint8_t) (0x01 << 1))

  Auto-wakeup asynchronous prescaler divider [1] (in _AWU_APR)
- #define _AWU_APRE2 ((uint8_t) (0x01 << 2))

  Auto-wakeup asynchronous prescaler divider [2] (in _AWU_APR)
- #define _AWU_APRE3 ((uint8_t) (0x01 << 3))

  Auto-wakeup asynchronous prescaler divider [3] (in _AWU_APR)
- #define _AWU_APRE4 ((uint8_t) (0x01 << 4))

  Auto-wakeup asynchronous prescaler divider [4] (in _AWU_APR)
- #define _AWU_APRE5 ((uint8_t) (0x01 << 5))

  Auto-wakeup asynchronous prescaler divider [5] (in _AWU_APR)
- #define _AWU_AWUTB ((uint8_t) (0x0F << 0))

  Auto-wakeup timebase selection [3:0] (in _AWU_APR)
- #define _AWU_AWUTB0 ((uint8_t) (0x01 << 0))

  Auto-wakeup timebase selection [0] (in _AWU_APR)
- #define _AWU_AWUTB1 ((uint8_t) (0x01 << 1))

  Auto-wakeup timebase selection [1] (in _AWU_APR)
- #define _AWU_AWUTB2 ((uint8_t) (0x01 << 2))

  Auto-wakeup timebase selection [2] (in _AWU_APR)
- #define _AWU_AWUTB3 ((uint8_t) (0x01 << 3))

  Auto-wakeup timebase selection [3] (in _AWU_APR)
- #define _BEEP _SFR(BEEP_t, BEEP_AddressBase)

  Beeper struct/bit access.
- #define _BEEP_CSR _SFR(uint8_t, BEEP_AddressBase+0x00)

  Beeper control/status register (BEEP_CSR)
- #define _BEEP_CSR_RESET_VALUE ((uint8_t) 0x1F)

  Beeper control/status register reset value.
- #define _BEEP_BEEPDIV ((uint8_t) (0x1F << 0))

  Beeper clock prescaler divider [4:0] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV0 ((uint8_t) (0x01 << 0))

  Beeper clock prescaler divider [0] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV1 ((uint8_t) (0x01 << 1))

  Beeper clock prescaler divider [1] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV2 ((uint8_t) (0x01 << 2))

  Beeper clock prescaler divider [2] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV3 ((uint8_t) (0x01 << 3))

  Beeper clock prescaler divider [3] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV4 ((uint8_t) (0x01 << 4))

  Beeper clock prescaler divider [4] (in _BEEP_CSR)
- #define _BEEP_BEEPEN ((uint8_t) (0x01 << 5))

  Beeper enable [0] (in _BEEP_CSR)

- #define _BEEP_BEEPSEL ((uint8_t) (0x03 << 6))
  Beeper frequency selection [1:0] (in _BEEP_CSR)
- #define _BEEP_BEEPSEL0 ((uint8_t) (0x01 << 6))
  Beeper frequency selection [0] (in _BEEP_CSR)
- #define _BEEP_BEEPSEL1 ((uint8_t) (0x01 << 7))
  Beeper frequency selection [1] (in _BEEP_CSR)
- #define _SPI _SFR(SPI_t, SPI_AddressBase)
  register for SPI control
- #define _SPI_CR1 _SFR(uint8_t, SPI_AddressBase+0x00)
  SPI control register 1.
- #define _SPI_CR2 _SFR(uint8_t, SPI_AddressBase+0x01)
  SPI control register 2.
- #define _SPI_ICR _SFR(uint8_t, SPI_AddressBase+0x02)
  SPI interrupt control register.
- #define _SPI_SR _SFR(uint8_t, SPI_AddressBase+0x03)
  SPI status register.
- #define _SPI_DR _SFR(uint8_t, SPI_AddressBase+0x04)
  SPI data register.
- #define _SPI_CR1_RESET_VALUE ((uint8_t) 0x00)
  SPI Control Register 1 reset value.
- #define _SPI_CR2_RESET_VALUE ((uint8_t) 0x00)
  SPI Control Register 2 reset value.
- #define _SPI_ICR_RESET_VALUE ((uint8_t) 0x00)
  SPI Interrupt Control Register reset value.
- #define _SPI_SR_RESET_VALUE ((uint8_t) 0x02)
  SPI Status Register reset value.
- #define _SPI_DR_RESET_VALUE ((uint8_t) 0x00)
  SPI Data Register reset value.
- #define _SPI_CPHA ((uint8_t) (0x01 << 0))
  SPI Clock phase [0] (in _SPI_CR1)
- #define _SPI_CPOL ((uint8_t) (0x01 << 1))
  SPI Clock polarity [0] (in _SPI_CR1)
- #define _SPI_MSTR ((uint8_t) (0x01 << 2))
  SPI Master/slave selection [0] (in _SPI_CR1)
- #define _SPI_BR ((uint8_t) (0x07 << 3))
  SPI Baudrate control [2:0] (in _SPI_CR1)
- #define _SPI_BR0 ((uint8_t) (0x01 << 3))
  SPI Baudrate control [0] (in _SPI_CR1)
- #define _SPI_BR1 ((uint8_t) (0x01 << 4))
  SPI Baudrate control [1] (in _SPI_CR1)
- #define _SPI_BR2 ((uint8_t) (0x01 << 5))
  SPI Baudrate control [2] (in _SPI_CR1)
- #define _SPI_SPE ((uint8_t) (0x01 << 6))
  SPI enable [0] (in _SPI_CR1)
- #define _SPI_LSBFIRST ((uint8_t) (0x01 << 7))
  SPI Frame format [0] (in _SPI_CR1)
- #define _SPI_SSI ((uint8_t) (0x01 << 0))
  SPI Internal slave select [0] (in _SPI_CR2)
- #define _SPI_SSM ((uint8_t) (0x01 << 1))
  SPI Software slave management [0] (in _SPI_CR2)
- #define _SPI_RXONLY ((uint8_t) (0x01 << 2))

SPI Receive only [0] (in _SPI_CR2)

- #define _SPI_BDOE ((uint8_t) (0x01 << 6))

  SPI Input/Output enable in bidirectional mode [0] (in _SPI_CR2)
- #define _SPI_BDM ((uint8_t) (0x01 << 7))

  SPI Bidirectional data mode enable [0] (in _SPI_CR2)
- #define _SPI_WKIE ((uint8_t) (0x01 << 4))

  SPI Wakeup interrupt enable [0] (in _SPI_ICR)
- #define _SPI_ERRIE ((uint8_t) (0x01 << 5))

  SPI Error interrupt enable [0] (in _SPI_ICR)
- #define _SPI_RXIE ((uint8_t) (0x01 << 6))

  SPI Rx buffer not empty interrupt enable [0] (in _SPI_ICR)
- #define _SPI_TXIE ((uint8_t) (0x01 << 7))

  SPI Tx buffer empty interrupt enable [0] (in _SPI_ICR)
- #define _SPI_RXNE ((uint8_t) (0x01 << 0))

  SPI Receive buffer not empty [0] (in _SPI_SR)
- #define _SPI_TXE ((uint8_t) (0x01 << 1))

  SPI Transmit buffer empty [0] (in _SPI_SR)
- #define _SPI_WKUP ((uint8_t) (0x01 << 3))

  SPI Wakeup flag [0] (in _SPI_SR)
- #define _SPI_MODF ((uint8_t) (0x01 << 5))

  SPI Mode fault [0] (in _SPI_SR)
- #define _SPI_OVR ((uint8_t) (0x01 << 6))

  SPI Overrun flag [0] (in _SPI_SR)
- #define _SPI_BSY ((uint8_t) (0x01 << 7))

  SPI Busy flag [0] (in _SPI_SR)
- #define _I2C _SFR(I2C_t, I2C_AddressBase)

  register for SPI control
- #define _I2C_CR1 _SFR(uint8_t, I2C_AddressBase+0x00)

  I2C Control register 1.
- #define _I2C_CR2 _SFR(uint8_t, I2C_AddressBase+0x01)

  I2C Control register 2.
- #define _I2C_FREQR _SFR(uint8_t, I2C_AddressBase+0x02)

  I2C Frequency register.
- #define _I2C_OARL _SFR(uint8_t, I2C_AddressBase+0x03)

  I2C own address register low byte.
- #define _I2C_OARH _SFR(uint8_t, I2C_AddressBase+0x04)

  I2C own address register high byte.
- #define _I2C_DR _SFR(uint8_t, I2C_AddressBase+0x06)

  I2C data register.
- #define _I2C_SR1 _SFR(uint8_t, I2C_AddressBase+0x07)

  I2C Status register 1.
- #define _I2C_SR2 _SFR(uint8_t, I2C_AddressBase+0x08)

  I2C Status register 2.
- #define _I2C_SR3 _SFR(uint8_t, I2C_AddressBase+0x09)

  I2C Status register 3.
- #define _I2C_ITR _SFR(uint8_t, I2C_AddressBase+0x0A)

  I2C Interrupt register.
- #define _I2C_CCRL _SFR(uint8_t, I2C_AddressBase+0x0B)

  I2C Clock control register low byte.
- #define _I2C_CCRH _SFR(uint8_t, I2C_AddressBase+0x0C)

  I2C Clock control register high byte.

- #define _I2C_TRISER _SFR(uint8_t, I2C_AddressBase+0x0D)
  I2C rise time register.
- #define _I2C_CR1_RESET_VALUE ((uint8_t) 0x00)
  I2C Control register 1 reset value.
- #define _I2C_CR2_RESET_VALUE ((uint8_t) 0x00)
  I2C Control register 2 reset value.
- #define _I2C_FREQR_RESET_VALUE ((uint8_t) 0x00)
  I2C Frequency register reset value.
- #define _I2C_OARL_RESET_VALUE ((uint8_t) 0x00)
  I2C own address register low byte reset value.
- #define _I2C_OARH_RESET_VALUE ((uint8_t) 0x00)
  I2C own address register high byte reset value.
- #define _I2C_DR_RESET_VALUE ((uint8_t) 0x00)
  I2C data register reset value.
- #define _I2C_SR1_RESET_VALUE ((uint8_t) 0x00)
  I2C Status register 1 reset value.
- #define _I2C_SR2_RESET_VALUE ((uint8_t) 0x00)
  I2C Status register 2 reset value.
- #define _I2C_SR3_RESET_VALUE ((uint8_t) 0x00)
  I2C Status register 3 reset value.
- #define _I2C_ITR_RESET_VALUE ((uint8_t) 0x00)
  I2C Interrupt register reset value.
- #define _I2C_CCRL_RESET_VALUE ((uint8_t) 0x00)
  I2C Clock control register low byte reset value.
- #define _I2C_CCRH_RESET_VALUE ((uint8_t) 0x00)
  I2C Clock control register high byte reset value.
- #define _I2C_TRISER_RESET_VALUE ((uint8_t) 0x02)
  I2C rise time register reset value.
- #define _I2C_PE ((uint8_t) (0x01 << 0))
  I2C Peripheral enable [0] (in _I2C_CR1)
- #define _I2C_ENGC ((uint8_t) (0x01 << 6))
  I2C General call enable [0] (in _I2C_CR1)
- #define _I2C_NOSTRETCH ((uint8_t) (0x01 << 7))
  I2C Clock stretching disable (Slave mode) [0] (in _I2C_CR1)
- #define _I2C_START ((uint8_t) (0x01 << 0))
  I2C Start generation [0] (in _I2C_CR2)
- #define _I2C_STOP ((uint8_t) (0x01 << 1))
  I2C Stop generation [0] (in _I2C_CR2)
- #define _I2C_ACK ((uint8_t) (0x01 << 2))
  I2C Acknowledge enable [0] (in _I2C_CR2)
- #define _I2C_POS ((uint8_t) (0x01 << 3))
  I2C Acknowledge position (for data reception) [0] (in _I2C_CR2)
- #define _I2C_SWRST ((uint8_t) (0x01 << 7))
  I2C Software reset [0] (in _I2C_CR2)
- #define _I2C_FREQ ((uint8_t) (0x3F << 0))
  I2C Peripheral clock frequency [5:0] (in _I2C_FREQR)
- #define _I2C_FREQ0 ((uint8_t) (0x01 << 0))
  I2C Peripheral clock frequency [0] (in _I2C_FREQR)
- #define _I2C_FREQ1 ((uint8_t) (0x01 << 1))
  I2C Peripheral clock frequency [1] (in _I2C_FREQR)
- #define _I2C_FREQ2 ((uint8_t) (0x01 << 2))

I2C Peripheral clock frequency [2] (in _I2C_FREQR)

- #define _I2C_FREQ3 ((uint8_t) (0x01 << 3))

  I2C Peripheral clock frequency [3] (in _I2C_FREQR)
- #define _I2C_FREQ4 ((uint8_t) (0x01 << 4))

  I2C Peripheral clock frequency [4] (in _I2C_FREQR)
- #define _I2C_FREQ5 ((uint8_t) (0x01 << 5))

  I2C Peripheral clock frequency [5] (in _I2C_FREQR)
- #define _I2C_ADD0 ((uint8_t) (0x01 << 0))

  I2C Interface address [0] (in 10-bit address mode) (in _I2C_OARL)
- #define _I2C_ADD1 ((uint8_t) (0x01 << 1))

  I2C Interface address [1] (in _I2C_OARL)
- #define _I2C_ADD2 ((uint8_t) (0x01 << 2))

  I2C Interface address [2] (in _I2C_OARL)
- #define _I2C_ADD3 ((uint8_t) (0x01 << 3))

  I2C Interface address [3] (in _I2C_OARL)
- #define _I2C_ADD4 ((uint8_t) (0x01 << 4))

  I2C Interface address [4] (in _I2C_OARL)
- #define _I2C_ADD5 ((uint8_t) (0x01 << 5))

  I2C Interface address [5] (in _I2C_OARL)
- #define _I2C_ADD6 ((uint8_t) (0x01 << 6))

  I2C Interface address [6] (in _I2C_OARL)
- #define _I2C_ADD7 ((uint8_t) (0x01 << 7))

  I2C Interface address [7] (in _I2C_OARL)
- #define _I2C_ADD_8_9 ((uint8_t) (0x03 << 1))

  I2C Interface address [9:8] (in 10-bit address mode) (in _I2C_OARH)
- #define _I2C_ADD8 ((uint8_t) (0x01 << 1))

  I2C Interface address [8] (in _I2C_OARH)
- #define _I2C_ADD9 ((uint8_t) (0x01 << 2))

  I2C Interface address [9] (in _I2C_OARH)
- #define _I2C_ADDCONF ((uint8_t) (0x01 << 6))

  I2C Address mode configuration [0] (in _I2C_OARH)
- #define _I2C_ADDMODE ((uint8_t) (0x01 << 7))

  I2C 7-/10-bit addressing mode (Slave mode) [0] (in _I2C_OARH)
- #define _I2C_SB ((uint8_t) (0x01 << 0))

  I2C Start bit (Master mode) [0] (in _I2C_SR1)
- #define _I2C_ADDR ((uint8_t) (0x01 << 1))

  I2C Address sent (Master mode) / matched (Slave mode) [0] (in _I2C_SR1)
- #define _I2C_BTF ((uint8_t) (0x01 << 2))

  I2C Byte transfer finished [0] (in _I2C_SR1)
- #define _I2C_ADD10 ((uint8_t) (0x01 << 3))

  I2C 10-bit header sent (Master mode) [0] (in _I2C_SR1)
- #define _I2C_STOPF ((uint8_t) (0x01 << 4))

  I2C Stop detection (Slave mode) [0] (in _I2C_SR1)
- #define _I2C_RXNE ((uint8_t) (0x01 << 6))

  I2C Data register not empty (receivers) [0] (in _I2C_SR1)
- #define _I2C_TXE ((uint8_t) (0x01 << 7))

  I2C Data register empty (transmitters) [0] (in _I2C_SR1)
- #define _I2C_BERR ((uint8_t) (0x01 << 0))

  I2C Bus error [0] (in _I2C_SR2)
- #define _I2C_ARLO ((uint8_t) (0x01 << 1))

  I2C Arbitration lost (Master mode) [0] (in _I2C_SR2)

- #define _I2C_AF ((uint8_t) (0x01 << 2))
  I2C Acknowledge failure [0] (in _I2C_SR2)
- #define _I2C_OVR ((uint8_t) (0x01 << 3))
  I2C Overrun/underrun [0] (in _I2C_SR2)
- #define _I2C_WUFH ((uint8_t) (0x01 << 5))
  I2C Wakeup from Halt [0] (in _I2C_SR2)
- #define _I2C_MSL ((uint8_t) (0x01 << 0))
  I2C Master/Slave [0] (in _I2C_SR3)
- #define _I2C_BUSY ((uint8_t) (0x01 << 1))
  I2C Bus busy [0] (in _I2C_SR3)
- #define _I2C_TRA ((uint8_t) (0x01 << 2))
  I2C Transmitter/Receiver [0] (in _I2C_SR3)
- #define _I2C_GENCALL ((uint8_t) (0x01 << 4))
  I2C General call header (Slave mode) [0] (in _I2C_SR3)
- #define _I2C_DUALF ((uint8_t) (0x01 << 7))
  Dual flag (Slave mode) [0] (in _I2C_SR3)
- #define _I2C_ITERREN ((uint8_t) (0x01 << 0))
  I2C Error interrupt enable [0] (in _I2C_ITR)
- #define _I2C_ITEVTEN ((uint8_t) (0x01 << 1))
  I2C Event interrupt enable [0] (in _I2C_ITR)
- #define _I2C_ITBUFEN ((uint8_t) (0x01 << 2))
  I2C Buffer interrupt enable [0] (in _I2C_ITR)
- #define _I2C_CCR ((uint8_t) (0x0F << 0))
  I2C Clock control register (Master mode) [3:0] (in _I2C_CCRH)
- #define _I2C_CCR0 ((uint8_t) (0x01 << 0))
  I2C Clock control register (Master mode) [0] (in _I2C_CCRH)
- #define _I2C_CCR1 ((uint8_t) (0x01 << 1))
  I2C Clock control register (Master mode) [1] (in _I2C_CCRH)
- #define _I2C_CCR2 ((uint8_t) (0x01 << 2))
  I2C Clock control register (Master mode) [2] (in _I2C_CCRH)
- #define _I2C_CCR3 ((uint8_t) (0x01 << 3))
  I2C Clock control register (Master mode) [3] (in _I2C_CCRH)
- #define _I2C_DUTY ((uint8_t) (0x01 << 6))
  I2C Fast mode duty cycle [0] (in _I2C_CCRH)
- #define _I2C_FS ((uint8_t) (0x01 << 7))
  I2C Master mode selection [0] (in _I2C_CCRH)
- #define _I2C_TRISE ((uint8_t) (0x3F << 0))
  I2C Maximum rise time (Master mode) [5:0] (in _I2C_TRISER)
- #define _I2C_TRISE0 ((uint8_t) (0x01 << 0))
  I2C Maximum rise time (Master mode) [0] (in _I2C_TRISER)
- #define _I2C_TRISE1 ((uint8_t) (0x01 << 1))
  I2C Maximum rise time (Master mode) [1] (in _I2C_TRISER)
- #define _I2C_TRISE2 ((uint8_t) (0x01 << 2))
  I2C Maximum rise time (Master mode) [2] (in _I2C_TRISER)
- #define _I2C_TRISE3 ((uint8_t) (0x01 << 3))
  I2C Maximum rise time (Master mode) [3] (in _I2C_TRISER)
- #define _I2C_TRISE4 ((uint8_t) (0x01 << 4))
  I2C Maximum rise time (Master mode) [4] (in _I2C_TRISER)
- #define _I2C_TRISE5 ((uint8_t) (0x01 << 5))
  I2C Maximum rise time (Master mode) [5] (in _I2C_TRISER)
- #define _USART _SFR(USART_t, USART_AddressBase)

USART struct/bit access.

- #define _USART_SR _SFR(uint8_t, USART_AddressBase+0x00)

  USART Status register.
- #define _USART_DR _SFR(uint8_t, USART_AddressBase+0x01)

  USART data register.
- #define _USART_BRR1 _SFR(uint8_t, USART_AddressBase+0x02)

  USART Baud rate register 1.
- #define _USART_BRR2 _SFR(uint8_t, USART_AddressBase+0x03)

  USART Baud rate register 2.
- #define _USART_CR1 _SFR(uint8_t, USART_AddressBase+0x04)

  USART Control register 1.
- #define _USART_CR2 _SFR(uint8_t, USART_AddressBase+0x05)

  USART Control register 2.
- #define _USART_CR3 _SFR(uint8_t, USART_AddressBase+0x06)

  USART Control register 3.
- #define _USART_CR4 _SFR(uint8_t, USART_AddressBase+0x07)

  USART Control register 4.
- #define _USART_SR_RESET_VALUE ((uint8_t) 0xC0)

  USART Status register reset value.
- #define _USART_BRR1_RESET_VALUE ((uint8_t) 0x00)

  USART Baud rate register 1 reset value.
- #define _USART_BRR2_RESET_VALUE ((uint8_t) 0x00)

  USART Baud rate register 2 reset value.
- #define _USART_CR1_RESET_VALUE ((uint8_t) 0x00)

  USART Control register 1 reset value.
- #define _USART_CR2_RESET_VALUE ((uint8_t) 0x00)

  USART Control register 2 reset value.
- #define _USART_CR3_RESET_VALUE ((uint8_t) 0x00)

  USART Control register 3 reset value.
- #define _USART_CR4_RESET_VALUE ((uint8_t) 0x00)

  USART Control register 4 reset value.
- #define _USART_PE ((uint8_t) (0x01 << 0))

  USART Parity error [0] (in _USART_SR)
- #define _USART_FE ((uint8_t) (0x01 << 1))

  USART Framing error [0] (in _USART_SR)
- #define _USART_NF ((uint8_t) (0x01 << 2))

  USART Noise flag [0] (in _USART_SR)
- #define _USART_OR_LHE ((uint8_t) (0x01 << 3))

  USART LIN Header Error (LIN Slave mode) / Overrun error [0] (in _USART_SR)
- #define _USART_IDLE ((uint8_t) (0x01 << 4))

  USART IDLE line detected [0] (in _USART_SR)
- #define _USART_RXNE ((uint8_t) (0x01 << 5))

  USART Read data register not empty [0] (in _USART_SR)
- #define _USART_TC ((uint8_t) (0x01 << 6))

  USART Transmission complete [0] (in _USART_SR)
- #define _USART_TXE ((uint8_t) (0x01 << 7))

  USART Transmit data register empty [0] (in _USART_SR)
- #define _USART_PIEN ((uint8_t) (0x01 << 0))

  USART Parity interrupt enable [0] (in _USART_CR1)
- #define _USART_PS ((uint8_t) (0x01 << 1))

  USART Parity selection [0] (in _USART_CR1)

- #define _USART_PCEN ((uint8_t) (0x01 << 2))
  USART Parity control enable [0] (in _USART_CR1)
- #define _USART_WAKE ((uint8_t) (0x01 << 3))
  USART Wakeup method [0] (in _USART_CR1)
- #define _USART_M ((uint8_t) (0x01 << 4))
  USART word length [0] (in _USART_CR1)
- #define _USART_UARTD ((uint8_t) (0x01 << 5))
  USART Disable (for low power consumption) [0] (in _USART_CR1)
- #define _USART_T8 ((uint8_t) (0x01 << 6))
  USART Transmit Data bit 8 (in 9-bit mode) [0] (in _USART_CR1)
- #define _USART_R8 ((uint8_t) (0x01 << 7))
  USART Receive Data bit 8 (in 9-bit mode) [0] (in _USART_CR1)
- #define _USART_SBK ((uint8_t) (0x01 << 0))
  USART Send break [0] (in _USART_CR2)
- #define _USART_RWU ((uint8_t) (0x01 << 1))
  USART Receiver wakeup [0] (in _USART_CR2)
- #define _USART_REN ((uint8_t) (0x01 << 2))
  USART Receiver enable [0] (in _USART_CR2)
- #define _USART_TEN ((uint8_t) (0x01 << 3))
  USART Transmitter enable [0] (in _USART_CR2)
- #define _USART_ILIEN ((uint8_t) (0x01 << 4))
  USART IDLE Line interrupt enable [0] (in _USART_CR2)
- #define _USART_RIEN ((uint8_t) (0x01 << 5))
  USART Receiver interrupt enable [0] (in _USART_CR2)
- #define _USART_TCIEN ((uint8_t) (0x01 << 6))
  USART Transmission complete interrupt enable [0] (in _USART_CR2)
- #define _USART_TIEN ((uint8_t) (0x01 << 7))
  USART Transmitter interrupt enable [0] (in _USART_CR2)
- #define _USART_LBCL ((uint8_t) (0x01 << 0))
  USART Last bit clock pulse [0] (in _USART_CR3)
- #define _USART_CPHA ((uint8_t) (0x01 << 1))
  USART Clock phase [0] (in _USART_CR3)
- #define _USART_CPOL ((uint8_t) (0x01 << 2))
  USART Clock polarity [0] (in _USART_CR3)
- #define _USART_CKEN ((uint8_t) (0x01 << 3))
  USART Clock enable [0] (in _USART_CR3)
- #define _USART_STOP ((uint8_t) (0x03 << 4))
  USART STOP bits [1:0] (in _USART_CR3)
- #define _USART_STOP0 ((uint8_t) (0x01 << 4))
  USART STOP bits [0] (in _USART_CR3)
- #define _USART_STOP1 ((uint8_t) (0x01 << 5))
  USART STOP bits [1] (in _USART_CR3)
- #define _USART_ADD ((uint8_t) (0x0F << 0))
  USART Address of the UART node [3:0] (in _USART_CR4)
- #define _USART_ADD0 ((uint8_t) (0x01 << 0))
  USART Address of the UART node [0] (in _USART_CR4)
- #define _USART_ADD1 ((uint8_t) (0x01 << 1))
  USART Address of the UART node [1] (in _USART_CR4)
- #define _USART_ADD2 ((uint8_t) (0x01 << 2))
  USART Address of the UART node [2] (in _USART_CR4)
- #define _USART_ADD3 ((uint8_t) (0x01 << 3))

USART Address of the UART node [3] (in _USART_CR4)

- #define _WFE _SFR(WFE_t, WFE_AddressBase)

  WFE struct/bit access.
- #define _WFE_CR1 _SFR(uint8_t, WFE_AddressBase+0x00)

  WFE Control register 1.
- #define _WFE_CR2 _SFR(uint8_t, WFE_AddressBase+0x01)

  WFE Control register 2.
- #define _WFE_CR1_RESET_VALUE ((uint8_t) 0x03)

  WFE Control register 1 reset value.
- #define _WFE_CR2_RESET_VALUE ((uint8_t) 0x00)

  WFE Control register 2 reset value.
- #define _WFE_TIM2_EV0 ((uint8_t) (0x01 << 0))

  TIM2 update, trigger or break event [0] (in _WFE_CR1)
- #define _WFE_TIM2_EV1 ((uint8_t) (0x01 << 1))

  TIM2 capture or compare event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV0 ((uint8_t) (0x01 << 4))

  Interrupt on pin 0 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV1 ((uint8_t) (0x01 << 5))

  Interrupt on pin 1 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV2 ((uint8_t) (0x01 << 6))

  Interrupt on pin 2 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV3 ((uint8_t) (0x01 << 7))

  Interrupt on pin 3 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV4 ((uint8_t) (0x01 << 0))

  Interrupt on pin 4 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV5 ((uint8_t) (0x01 << 1))

  Interrupt on pin 5 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV6 ((uint8_t) (0x01 << 2))

  Interrupt on pin 6 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV7 ((uint8_t) (0x01 << 3))

  Interrupt on pin 7 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EVB ((uint8_t) (0x01 << 4))

  Interrupt on port B event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EVD ((uint8_t) (0x01 << 5))

  Interrupt on port D event [0] (in _WFE_CR1)
- #define _TIM2 _SFR(TIM2_3_t, TIM2_AddressBase)

  TIM2 struct/bit access.
- #define _TIM2_CR1 _SFR(uint8_t, TIM2_AddressBase+0x00)

  TIM2 control register 1.
- #define _TIM2_CR2 _SFR(uint8_t, TIM2_AddressBase+0x01)

  TIM2 control register 2.
- #define _TIM2_SMCR _SFR(uint8_t, TIM2_AddressBase+0x02)

  TIM2 Slave mode control register.
- #define _TIM2_ETR _SFR(uint8_t, TIM2_AddressBase+0x03)

  TIM2 External trigger register.
- #define _TIM2_IER _SFR(uint8_t, TIM2_AddressBase+0x04)

  TIM2 interrupt enable register.
- #define _TIM2_SR1 _SFR(uint8_t, TIM2_AddressBase+0x05)

  TIM2 status register 1.
- #define _TIM2_SR2 _SFR(uint8_t, TIM2_AddressBase+0x06)

  TIM2 status register 2.

- #define _TIM2_EGR _SFR(uint8_t, TIM2_AddressBase+0x07)

  TIM2 Event generation register.
- #define _TIM2_CCMR1 _SFR(uint8_t, TIM2_AddressBase+0x08)

  TIM2 Capture/compare mode register 1.
- #define _TIM2_CCMR2 _SFR(uint8_t, TIM2_AddressBase+0x09)

  TIM2 Capture/compare mode register 2.
- #define _TIM2_CCER1 _SFR(uint8_t, TIM2_AddressBase+0x0A)

  TIM2 Capture/compare enable register 1.
- #define _TIM2_CNTRH _SFR(uint8_t, TIM2_AddressBase+0x0B)

  TIM2 counter register high byte.
- #define _TIM2_CNTRL _SFR(uint8_t, TIM2_AddressBase+0x0C)

  TIM2 counter register low byte.
- #define _TIM2_PSCR _SFR(uint8_t, TIM2_AddressBase+0x0D)

  TIM2 clock prescaler register.
- #define _TIM2_ARRH _SFR(uint8_t, TIM2_AddressBase+0x0E)

  TIM2 auto-reload register high byte.
- #define _TIM2_ARRL _SFR(uint8_t, TIM2_AddressBase+0x0F)

  TIM2 auto-reload register low byte.
- #define _TIM2_CCR1H _SFR(uint8_t, TIM2_AddressBase+0x10)

  TIM2 16-bit capture/compare value 1 high byte.
- #define _TIM2_CCR1L _SFR(uint8_t, TIM2_AddressBase+0x11)

  TIM2 16-bit capture/compare value 1 low byte.
- #define _TIM2_CCR2H _SFR(uint8_t, TIM2_AddressBase+0x12)

  TIM2 16-bit capture/compare value 2 high byte.
- #define _TIM2_CCR2L _SFR(uint8_t, TIM2_AddressBase+0x13)

  TIM2 16-bit capture/compare value 2 low byte.
- #define _TIM2_BKR _SFR(uint8_t, TIM2_AddressBase+0x14)

  TIM2 Break register.
- #define _TIM2_OISR _SFR(uint8_t, TIM2_AddressBase+0x15)

  TIM2 Output idle state register.
- #define _TIM2_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 control register 1 reset value.
- #define _TIM2_CR2_RESET_VALUE ((uint8_t) 0x00)

  TIM2 control register 2 reset value.
- #define _TIM2_SMCR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Slave mode control register reset value.
- #define _TIM2_ETR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 External trigger register reset value.
- #define _TIM2_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM2 interrupt enable register reset value.
- #define _TIM2_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 status register 1 reset value.
- #define _TIM2_SR2_RESET_VALUE ((uint8_t) 0x00)

  TIM2 status register 2 reset value.
- #define _TIM2_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Event generation register reset value.
- #define _TIM2_CCMR1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare mode register 1 reset value.
- #define _TIM2_CCMR2_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare mode register 2 reset value.
- #define _TIM2_CCER1_RESET_VALUE ((uint8_t) 0x00)

TIM2 Capture/compare enable register 1 reset value.

- #define _TIM2_CNTRH_RESET_VALUE ((uint8_t) 0x00)

  TIM2 counter register high byte reset value.
- #define _TIM2_CNTRL_RESET_VALUE ((uint8_t) 0x00)

  TIM2 counter register low byte reset value.
- #define _TIM2_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 clock prescaler register reset value.
- #define _TIM2_ARRH_RESET_VALUE ((uint8_t) 0xFF)

  TIM2 auto-reload register high byte reset value.
- #define _TIM2_ARRL_RESET_VALUE ((uint8_t) 0xFF)

  TIM2 auto-reload register low byte reset value.
- #define _TIM2_RCR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Repetition counter reset value.
- #define _TIM2_CCR1H_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 1 high byte reset value.
- #define _TIM2_CCR1L_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 1 low byte reset value.
- #define _TIM2_CCR2H_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 2 high byte reset value.
- #define _TIM2_CCR2L_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 2 low byte reset value.
- #define _TIM2_BKR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Break register reset value.
- #define _TIM2_OISR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Output idle state register reset value.
- #define _TIM2_CEN ((uint8_t) (0x01 << 0))

  TIM2 Counter enable [0] (in _TIM2_CR1)
- #define _TIM2_UDIS ((uint8_t) (0x01 << 1))

  TIM2 Update disable [0] (in _TIM2_CR1)
- #define _TIM2_URS ((uint8_t) (0x01 << 2))

  TIM2 Update request source [0] (in _TIM2_CR1)
- #define _TIM2_OPM ((uint8_t) (0x01 << 3))

  TIM2 One-pulse mode [0] (in _TIM2_CR1)
- #define _TIM2_DIR ((uint8_t) (0x01 << 4))

  TIM2 Direction [0] (in _TIM2_CR1)
- #define _TIM2_ARPE ((uint8_t) (0x01 << 7))

  TIM2 Auto-reload preload enable [0] (in _TIM2_CR1)
- #define _TIM2_MMS ((uint8_t) (0x07 << 4))

  TIM2 Master mode selection [2:0] (in _TIM2_CR2)
- #define _TIM2_MMS0 ((uint8_t) (0x01 << 4))

  TIM2 Master mode selection [0] (in _TIM2_CR2)
- #define _TIM2_MMS1 ((uint8_t) (0x01 << 5))

  TIM2 Master mode selection [1] (in _TIM2_CR2)
- #define _TIM2_MMS2 ((uint8_t) (0x01 << 6))

  TIM2 Master mode selection [2] (in _TIM2_CR2)
- #define _TIM2_SMS ((uint8_t) (0x07 << 0))

  TIM2 Clock/trigger/slave mode selection [2:0] (in _TIM2_SMCR)
- #define _TIM2_SMS0 ((uint8_t) (0x01 << 0))

  TIM2 Clock/trigger/slave mode selection [0] (in _TIM2_SMCR)
- #define _TIM2_SMS1 ((uint8_t) (0x01 << 1))

  TIM2 Clock/trigger/slave mode selection [1] (in _TIM2_SMCR)

- #define _TIM2_SMS2 ((uint8_t) (0x01 << 2))
  TIM2 Clock/trigger/slave mode selection [2] (in _TIM2_SMCR)
- #define _TIM2_TS ((uint8_t) (0x07 << 4))
  TIM2 Trigger selection [2:0] (in _TIM2_SMCR)
- #define _TIM2_TS0 ((uint8_t) (0x01 << 4))
  TIM2 Trigger selection [0] (in _TIM2_SMCR)
- #define _TIM2_TS1 ((uint8_t) (0x01 << 5))
  TIM2 Trigger selection [1] (in _TIM2_SMCR)
- #define _TIM2_TS2 ((uint8_t) (0x01 << 6))
  TIM2 Trigger selection [2] (in _TIM2_SMCR)
- #define _TIM2_MSM ((uint8_t) (0x01 << 7))
  TIM2 Master/slave mode [0] (in _TIM2_SMCR)
- #define _TIM2_ETF ((uint8_t) (0x0F << 0))
  TIM2 External trigger filter [3:0] (in _TIM2_ETR)
- #define _TIM2_ETF0 ((uint8_t) (0x01 << 0))
  TIM2 External trigger filter [0] (in _TIM2_ETR)
- #define _TIM2_ETF1 ((uint8_t) (0x01 << 1))
  TIM2 External trigger filter [1] (in _TIM2_ETR)
- #define _TIM2_ETF2 ((uint8_t) (0x01 << 2))
  TIM2 External trigger filter [2] (in _TIM2_ETR)
- #define _TIM2_ETF3 ((uint8_t) (0x01 << 3))
  TIM2 External trigger filter [3] (in _TIM2_ETR)
- #define _TIM2_ETPS ((uint8_t) (0x03 << 4))
  TIM2 External trigger prescaler [1:0] (in _TIM2_ETR)
- #define _TIM2_ETPS0 ((uint8_t) (0x01 << 4))
  TIM2 External trigger prescaler [0] (in _TIM2_ETR)
- #define _TIM2_ETPS1 ((uint8_t) (0x01 << 5))
  TIM2 External trigger prescaler [1] (in _TIM2_ETR)
- #define _TIM2_ECE ((uint8_t) (0x01 << 6))
  TIM2 External clock enable [0] (in _TIM2_ETR)
- #define _TIM2_ETP ((uint8_t) (0x01 << 7))
  TIM2 External trigger polarity [0] (in _TIM2_ETR)
- #define _TIM2_UIE ((uint8_t) (0x01 << 0))
  TIM2 Update interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_CC1IE ((uint8_t) (0x01 << 1))
  TIM2 Capture/compare 1 interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_CC2IE ((uint8_t) (0x01 << 2))
  TIM2 Capture/compare 2 interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_TIE ((uint8_t) (0x01 << 6))
  TIM2 Trigger interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_BIE ((uint8_t) (0x01 << 7))
  TIM2 Break interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_UIF ((uint8_t) (0x01 << 0))
  TIM2 Update interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC1IF ((uint8_t) (0x01 << 1))
  TIM2 Capture/compare 1 interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC2IF ((uint8_t) (0x01 << 2))
  TIM2 Capture/compare 2 interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_TIF ((uint8_t) (0x01 << 6))
  TIM2 Trigger interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_BIF ((uint8_t) (0x01 << 7))

TIM2 Break interrupt flag [0] (in _TIM2_SR1)

- #define _TIM2_CC1OF ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 overcapture flag [0] (in _TIM2_SR2)
- #define _TIM2_CC2OF ((uint8_t) (0x01 << 2))

  TIM2 Capture/compare 2 overcapture flag [0] (in _TIM2_SR2)
- #define _TIM2_UG ((uint8_t) (0x01 << 0))

  TIM2 Update generation [0] (in _TIM2_EGR)
- #define _TIM2_CC1G ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 generation [0] (in _TIM2_EGR)
- #define _TIM2_CC2G ((uint8_t) (0x01 << 2))

  TIM2 Capture/compare 2 generation [0] (in _TIM2_EGR)
- #define _TIM2_TG ((uint8_t) (0x01 << 6))

  TIM2 Trigger generation [0] (in _TIM2_EGR)
- #define _TIM2_BG ((uint8_t) (0x01 << 7))

  TIM2 Break generation [0] (in _TIM2_EGR)
- #define _TIM2_CC1S ((uint8_t) (0x03 << 0))

  TIM2 Compare 1 selection [1:0] (in _TIM2_CCMR1)
- #define _TIM2_CC1S0 ((uint8_t) (0x01 << 0))

  TIM2 Compare 1 selection [0] (in _TIM2_CCMR1)
- #define _TIM2_CC1S1 ((uint8_t) (0x01 << 1))

  TIM2 Compare 1 selection [1] (in _TIM2_CCMR1)
- #define _TIM2_OC1FE ((uint8_t) (0x01 << 2))

  TIM2 Output compare 1 fast enable [0] (in _TIM2_CCMR1)
- #define _TIM2_OC1PE ((uint8_t) (0x01 << 3))

  TIM2 Output compare 1 preload enable [0] (in _TIM2_CCMR1)
- #define _TIM2_OC1M ((uint8_t) (0x07 << 4))

  TIM2 Output compare 1 mode [2:0] (in _TIM2_CCMR1)
- #define _TIM2_OC1M0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 1 mode [0] (in _TIM2_CCMR1)
- #define _TIM2_OC1M1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 1 mode [1] (in _TIM2_CCMR1)
- #define _TIM2_OC1M2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 1 mode [2] (in _TIM2_CCMR1)
- #define _TIM2_IC1PSC ((uint8_t) (0x03 << 2))

  TIM2 Input capture 1 prescaler [1:0] (in _TIM2_CCMR1)
- #define _TIM2_IC1PSC0 ((uint8_t) (0x01 << 2))

  TIM2 Input capture 1 prescaler [0] (in _TIM2_CCMR1)
- #define _TIM2_IC1PSC1 ((uint8_t) (0x01 << 3))

  TIM2 Input capture 1 prescaler [1] (in _TIM2_CCMR1)
- #define _TIM2_IC1F ((uint8_t) (0x0F << 4))

  TIM2 Output compare 1 mode [3:0] (in _TIM2_CCMR1)
- #define _TIM2_IC1F0 ((uint8_t) (0x01 << 4))

  TIM2 Input capture 1 filter [0] (in _TIM2_CCMR1)
- #define _TIM2_IC1F1 ((uint8_t) (0x01 << 5))

  TIM2 Input capture 1 filter [1] (in _TIM2_CCMR1)
- #define _TIM2_IC1F2 ((uint8_t) (0x01 << 6))

  TIM2 Input capture 1 filter [2] (in _TIM2_CCMR1)
- #define _TIM2_IC1F3 ((uint8_t) (0x01 << 7))

  TIM2 Input capture 1 filter [3] (in _TIM2_CCMR1)
- #define _TIM2_CC2S ((uint8_t) (0x03 << 0))

  TIM2 Compare 2 selection [1:0] (in _TIM2_CCMR2)

- #define _TIM2_CC2S0 ((uint8_t) (0x01 << 0))
  TIM2 Compare 2 selection [0] (in _TIM2_CCMR2)
- #define _TIM2_CC2S1 ((uint8_t) (0x01 << 1))
  TIM2 Compare 2 selection [1] (in _TIM2_CCMR2)
- #define _TIM2_OC2FE ((uint8_t) (0x01 << 2))
  TIM2 Output compare 2 fast enable [0] (in _TIM2_CCMR2)
- #define _TIM2_OC2PE ((uint8_t) (0x01 << 3))
  TIM2 Output compare 2 preload enable [0] (in _TIM2_CCMR2)
- #define _TIM2_OC2M ((uint8_t) (0x07 << 4))
  TIM2 Output compare 2 mode [2:0] (in _TIM2_CCMR2)
- #define _TIM2_OC2M0 ((uint8_t) (0x01 << 4))
  TIM2 Output compare 2 mode [0] (in _TIM2_CCMR2)
- #define _TIM2_OC2M1 ((uint8_t) (0x01 << 5))
  TIM2 Output compare 2 mode [1] (in _TIM2_CCMR2)
- #define _TIM2_OC2M2 ((uint8_t) (0x01 << 6))
  TIM2 Output compare 2 mode [2] (in _TIM2_CCMR2)
- #define _TIM2_IC2PSC ((uint8_t) (0x03 << 2))
  TIM2 Input capture 2 prescaler [1:0] (in _TIM2_CCMR2)
- #define _TIM2_IC2PSC0 ((uint8_t) (0x01 << 2))
  TIM2 Input capture 2 prescaler [0] (in _TIM2_CCMR2)
- #define _TIM2_IC2PSC1 ((uint8_t) (0x01 << 3))
  TIM2 Input capture 2 prescaler [1] (in _TIM2_CCMR2)
- #define _TIM2_IC2F ((uint8_t) (0x0F << 4))
  TIM2 Output compare 2 mode [3:0] (in _TIM2_CCMR2)
- #define _TIM2_IC2F0 ((uint8_t) (0x01 << 4))
  TIM2 Input capture 2 filter [0] (in _TIM2_CCMR2)
- #define _TIM2_IC2F1 ((uint8_t) (0x01 << 5))
  TIM2 Input capture 2 filter [1] (in _TIM2_CCMR2)
- #define _TIM2_IC2F2 ((uint8_t) (0x01 << 6))
  TIM2 Input capture 2 filter [2] (in _TIM2_CCMR2)
- #define _TIM2_IC2F3 ((uint8_t) (0x01 << 7))
  TIM2 Input capture 2 filter [3] (in _TIM2_CCMR2)
- #define _TIM2_CC1E ((uint8_t) (0x01 << 0))
  TIM2 Capture/compare 1 output enable [0] (in _TIM2_CCER1)
- #define _TIM2_CC1P ((uint8_t) (0x01 << 1))
  TIM2 Capture/compare 1 output polarity [0] (in _TIM2_CCER1)
- #define _TIM2_CC2E ((uint8_t) (0x01 << 4))
  TIM2 Capture/compare 2 output enable [0] (in _TIM2_CCER1)
- #define _TIM2_CC2P ((uint8_t) (0x01 << 5))
  TIM2 Capture/compare 2 output polarity [0] (in _TIM2_CCER1)
- #define _TIM2_PSC ((uint8_t) (0x07 << 0))
  TIM2 prescaler [2:0] (in _TIM2_PSCR)
- #define _TIM2_PSC0 ((uint8_t) (0x01 << 0))
  TIM2 prescaler [0] (in _TIM2_PSCR)
- #define _TIM2_PSC1 ((uint8_t) (0x01 << 1))
  TIM2 prescaler [1] (in _TIM2_PSCR)
- #define _TIM2_PSC2 ((uint8_t) (0x01 << 2))
  TIM2 prescaler [2] (in _TIM2_PSCR)
- #define _TIM2_LOCK ((int8_t) (0x03 << 0))
  TIM2 Lock configuration [1:0] (in _TIM2_BKR)
- #define _TIM2_LOCK0 ((uint8_t) (0x01 << 0))

TIM2 Lock configuration [0] (in _TIM2_BKR)

- #define _TIM2_LOCK1 ((uint8_t) (0x01 << 1))

  TIM2 Lock configuration [1] (in _TIM2_BKR)
- #define _TIM2_OSSI ((uint8_t) (0x01 << 2))

  TIM2 Off state selection for idle mode [0] (in _TIM2_BKR)
- #define _TIM2_BKE ((uint8_t) (0x01 << 4))

  TIM2 Break enable [0] (in _TIM2_BKR)
- #define _TIM2_BKP ((uint8_t) (0x01 << 5))

  TIM2 Break polarity [0] (in _TIM2_BKR)
- #define _TIM2_AOE ((uint8_t) (0x01 << 6))

  TIM2 Automatic output enable [0] (in _TIM2_BKR)
- #define _TIM2_MOE ((uint8_t) (0x01 << 7))

  TIM2 Main output enable [0] (in _TIM2_BKR)
- #define _TIM2_OIS1 ((uint8_t) (0x01 << 0))

  TIM2 Output idle state 1 (OC1 output) [0] (in _TIM2_OISR)
- #define _TIM2_OIS2 ((uint8_t) (0x01 << 2))

  TIM2 Output idle state 2 (OC2 output) [0] (in _TIM2_OISR)
- #define _TIM3 _SFR(TIM3_3_t, TIM3_AddressBase)

  TIM3 struct/bit access.
- #define _TIM3_CR1 _SFR(uint8_t, TIM3_AddressBase+0x00)

  TIM3 control register 1.
- #define _TIM3_CR2 _SFR(uint8_t, TIM3_AddressBase+0x01)

  TIM3 control register 2.
- #define _TIM3_SMCR _SFR(uint8_t, TIM3_AddressBase+0x02)

  TIM3 Slave mode control register.
- #define _TIM3_ETR _SFR(uint8_t, TIM3_AddressBase+0x03)

  TIM3 External trigger register.
- #define _TIM3_IER _SFR(uint8_t, TIM3_AddressBase+0x04)

  TIM3 interrupt enable register.
- #define _TIM3_SR1 _SFR(uint8_t, TIM3_AddressBase+0x05)

  TIM3 status register 1.
- #define _TIM3_SR2 _SFR(uint8_t, TIM3_AddressBase+0x06)

  TIM3 status register 2.
- #define _TIM3_EGR _SFR(uint8_t, TIM3_AddressBase+0x07)

  TIM3 Event generation register.
- #define _TIM3_CCMR1 _SFR(uint8_t, TIM3_AddressBase+0x08)

  TIM3 Capture/compare mode register 1.
- #define _TIM3_CCMR2 _SFR(uint8_t, TIM3_AddressBase+0x09)

  TIM3 Capture/compare mode register 2.
- #define _TIM3_CCER1 _SFR(uint8_t, TIM3_AddressBase+0x0A)

  TIM3 Capture/compare enable register 1.
- #define _TIM3_CNTRH _SFR(uint8_t, TIM3_AddressBase+0x0B)

  TIM3 counter register high byte.
- #define _TIM3_CNTRL _SFR(uint8_t, TIM3_AddressBase+0x0C)

  TIM3 counter register low byte.
- #define _TIM3_PSCR _SFR(uint8_t, TIM3_AddressBase+0x0D)

  TIM3 clock prescaler register.
- #define _TIM3_ARRH _SFR(uint8_t, TIM3_AddressBase+0x0E)

  TIM3 auto-reload register high byte.
- #define _TIM3_ARRL _SFR(uint8_t, TIM3_AddressBase+0x0F)

  TIM3 auto-reload register low byte.

- #define _TIM3_CCR1H _SFR(uint8_t, TIM3_AddressBase+0x10)

  TIM3 16-bit capture/compare value 1 high byte.
- #define _TIM3_CCR1L _SFR(uint8_t, TIM3_AddressBase+0x11)

  TIM3 16-bit capture/compare value 1 low byte.
- #define _TIM3_CCR2H _SFR(uint8_t, TIM3_AddressBase+0x12)

  TIM3 16-bit capture/compare value 2 high byte.
- #define _TIM3_CCR2L _SFR(uint8_t, TIM3_AddressBase+0x13)

  TIM3 16-bit capture/compare value 2 low byte.
- #define _TIM3_BKR _SFR(uint8_t, TIM3_AddressBase+0x14)

  TIM3 Break register.
- #define _TIM3_OISR _SFR(uint8_t, TIM3_AddressBase+0x15)

  TIM3 Output idle state register.
- #define _TIM3_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 control register 1 reset value.
- #define _TIM3_CR2_RESET_VALUE ((uint8_t) 0x00)

  TIM3 control register 2 reset value.
- #define _TIM3_SMCR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Slave mode control register reset value.
- #define _TIM3_ETR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 External trigger register reset value.
- #define _TIM3_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM3 interrupt enable register reset value.
- #define _TIM3_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 status register 1 reset value.
- #define _TIM3_SR2_RESET_VALUE ((uint8_t) 0x00)

  TIM3 status register 2 reset value.
- #define _TIM3_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Event generation register reset value.
- #define _TIM3_CCMR1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Capture/compare mode register 1 reset value.
- #define _TIM3_CCMR2_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Capture/compare mode register 2 reset value.
- #define _TIM3_CCER1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Capture/compare enable register 1 reset value.
- #define _TIM3_CNTRH_RESET_VALUE ((uint8_t) 0x00)

  TIM3 counter register high byte reset value.
- #define _TIM3_CNTRL_RESET_VALUE ((uint8_t) 0x00)

  TIM3 counter register low byte reset value.
- #define _TIM3_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 clock prescaler register reset value.
- #define _TIM3_ARRH_RESET_VALUE ((uint8_t) 0xFF)

  TIM3 auto-reload register high byte reset value.
- #define _TIM3_ARRL_RESET_VALUE ((uint8_t) 0xFF)

  TIM3 auto-reload register low byte reset value.
- #define _TIM3_RCR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Repetition counter reset value.
- #define _TIM3_CCR1H_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 1 high byte reset value.
- #define _TIM3_CCR1L_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 1 low byte reset value.
- #define _TIM3_CCR2H_RESET_VALUE ((uint8_t) 0x00)

TIM3 16-bit capture/compare value 2 high byte reset value.

- #define _TIM3_CCR2L_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 2 low byte reset value.
- #define _TIM3_BKR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Break register reset value.
- #define _TIM3_OISR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Output idle state register reset value.
- #define _TIM3_CEN ((uint8_t) (0x01 << 0))

  TIM3 Counter enable [0] (in _TIM3_CR1)
- #define _TIM3_UDIS ((uint8_t) (0x01 << 1))

  TIM3 Update disable [0] (in _TIM3_CR1)
- #define _TIM3_URS ((uint8_t) (0x01 << 2))

  TIM3 Update request source [0] (in _TIM3_CR1)
- #define _TIM3_OPM ((uint8_t) (0x01 << 3))

  TIM3 One-pulse mode [0] (in _TIM3_CR1)
- #define _TIM3_DIR ((uint8_t) (0x01 << 4))

  TIM3 Direction [0] (in _TIM3_CR1)
- #define _TIM3_ARPE ((uint8_t) (0x01 << 7))

  TIM3 Auto-reload preload enable [0] (in _TIM3_CR1)
- #define _TIM3_MMS ((uint8_t) (0x07 << 4))

  TIM3 Master mode selection [2:0] (in _TIM3_CR2)
- #define _TIM3_MMS0 ((uint8_t) (0x01 << 4))

  TIM3 Master mode selection [0] (in _TIM3_CR2)
- #define _TIM3_MMS1 ((uint8_t) (0x01 << 5))

  TIM3 Master mode selection [1] (in _TIM3_CR2)
- #define _TIM3_MMS2 ((uint8_t) (0x01 << 6))

  TIM3 Master mode selection [2] (in _TIM3_CR2)
- #define _TIM3_SMS ((uint8_t) (0x07 << 0))

  TIM3 Clock/trigger/slave mode selection [2:0] (in _TIM3_SMCR)
- #define _TIM3_SMS0 ((uint8_t) (0x01 << 0))

  TIM3 Clock/trigger/slave mode selection [0] (in _TIM3_SMCR)
- #define _TIM3_SMS1 ((uint8_t) (0x01 << 1))

  TIM3 Clock/trigger/slave mode selection [1] (in _TIM3_SMCR)
- #define _TIM3_SMS2 ((uint8_t) (0x01 << 2))

  TIM3 Clock/trigger/slave mode selection [2] (in _TIM3_SMCR)
- #define _TIM3_TS ((uint8_t) (0x07 << 4))

  TIM3 Trigger selection [2:0] (in _TIM3_SMCR)
- #define _TIM3_TS0 ((uint8_t) (0x01 << 4))

  TIM3 Trigger selection [0] (in _TIM3_SMCR)
- #define _TIM3_TS1 ((uint8_t) (0x01 << 5))

  TIM3 Trigger selection [1] (in _TIM3_SMCR)
- #define _TIM3_TS2 ((uint8_t) (0x01 << 6))

  TIM3 Trigger selection [2] (in _TIM3_SMCR)
- #define _TIM3_MSM ((uint8_t) (0x01 << 7))

  TIM3 Master/slave mode [0] (in _TIM3_SMCR)
- #define _TIM3_ETF ((uint8_t) (0x0F << 0))

  TIM3 External trigger filter [3:0] (in _TIM3_ETR)
- #define _TIM3_ETF0 ((uint8_t) (0x01 << 0))

  TIM3 External trigger filter [0] (in _TIM3_ETR)
- #define _TIM3_ETF1 ((uint8_t) (0x01 << 1))

  TIM3 External trigger filter [1] (in _TIM3_ETR)

- #define _TIM3_ETF2 ((uint8_t) (0x01 << 2))
  TIM3 External trigger filter [2] (in _TIM3_ETR)
- #define _TIM3_ETF3 ((uint8_t) (0x01 << 3))
  TIM3 External trigger filter [3] (in _TIM3_ETR)
- #define _TIM3_ETPS ((uint8_t) (0x03 << 4))
  TIM3 External trigger prescaler [1:0] (in _TIM3_ETR)
- #define _TIM3_ETPS0 ((uint8_t) (0x01 << 4))
  TIM3 External trigger prescaler [0] (in _TIM3_ETR)
- #define _TIM3_ETPS1 ((uint8_t) (0x01 << 5))
  TIM3 External trigger prescaler [1] (in _TIM3_ETR)
- #define _TIM3_ECE ((uint8_t) (0x01 << 6))
  TIM3 External clock enable [0] (in _TIM3_ETR)
- #define _TIM3_ETP ((uint8_t) (0x01 << 7))
  TIM3 External trigger polarity [0] (in _TIM3_ETR)
- #define _TIM3_UIE ((uint8_t) (0x01 << 0))
  TIM3 Update interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_CC1IE ((uint8_t) (0x01 << 1))
  TIM3 Capture/compare 1 interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_CC2IE ((uint8_t) (0x01 << 2))
  TIM3 Capture/compare 2 interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_TIE ((uint8_t) (0x01 << 6))
  TIM3 Trigger interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_BIE ((uint8_t) (0x01 << 7))
  TIM3 Break interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_UIF ((uint8_t) (0x01 << 0))
  TIM3 Update interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_CC1IF ((uint8_t) (0x01 << 1))
  TIM3 Capture/compare 1 interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_CC2IF ((uint8_t) (0x01 << 2))
  TIM3 Capture/compare 2 interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_TIF ((uint8_t) (0x01 << 6))
  TIM3 Trigger interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_BIF ((uint8_t) (0x01 << 7))
  TIM3 Break interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_CC1OF ((uint8_t) (0x01 << 1))
  TIM3 Capture/compare 1 overcapture flag [0] (in _TIM3_SR2)
- #define _TIM3_CC2OF ((uint8_t) (0x01 << 2))
  TIM3 Capture/compare 2 overcapture flag [0] (in _TIM3_SR2)
- #define _TIM3_UG ((uint8_t) (0x01 << 0))
  TIM3 Update generation [0] (in _TIM3_EGR)
- #define _TIM3_CC1G ((uint8_t) (0x01 << 1))
  TIM3 Capture/compare 1 generation [0] (in _TIM3_EGR)
- #define _TIM3_CC2G ((uint8_t) (0x01 << 2))
  TIM3 Capture/compare 2 generation [0] (in _TIM3_EGR)
- #define _TIM3_TG ((uint8_t) (0x01 << 6))
  TIM3 Trigger generation [0] (in _TIM3_EGR)
- #define _TIM3_BG ((uint8_t) (0x01 << 7))
  TIM3 Break generation [0] (in _TIM3_EGR)
- #define _TIM3_CC1S ((uint8_t) (0x03 << 0))
  TIM3 Compare 1 selection [1:0] (in _TIM3_CCMR1)
- #define _TIM3_CC1S0 ((uint8_t) (0x01 << 0))

TIM3 Compare 1 selection [0] (in _TIM3_CCMR1)

- #define _TIM3_CC1S1 ((uint8_t) (0x01 << 1))

  TIM3 Compare 1 selection [1] (in _TIM3_CCMR1)
- #define _TIM3_OC1FE ((uint8_t) (0x01 << 2))

  TIM3 Output compare 1 fast enable [0] (in _TIM3_CCMR1)
- #define _TIM3_OC1PE ((uint8_t) (0x01 << 3))

  TIM3 Output compare 1 preload enable [0] (in _TIM3_CCMR1)
- #define _TIM3_OC1M ((uint8_t) (0x07 << 4))

  TIM3 Output compare 1 mode [2:0] (in _TIM3_CCMR1)
- #define _TIM3_OC1M0 ((uint8_t) (0x01 << 4))

  TIM3 Output compare 1 mode [0] (in _TIM3_CCMR1)
- #define _TIM3_OC1M1 ((uint8_t) (0x01 << 5))

  TIM3 Output compare 1 mode [1] (in _TIM3_CCMR1)
- #define _TIM3_OC1M2 ((uint8_t) (0x01 << 6))

  TIM3 Output compare 1 mode [2] (in _TIM3_CCMR1)
- #define _TIM3_IC1PSC ((uint8_t) (0x03 << 2))

  TIM3 Input capture 1 prescaler [1:0] (in _TIM3_CCMR1)
- #define _TIM3_IC1PSC0 ((uint8_t) (0x01 << 2))

  TIM3 Input capture 1 prescaler [0] (in _TIM3_CCMR1)
- #define _TIM3_IC1PSC1 ((uint8_t) (0x01 << 3))

  TIM3 Input capture 1 prescaler [1] (in _TIM3_CCMR1)
- #define _TIM3_IC1F ((uint8_t) (0x0F << 4))

  TIM3 Output compare 1 mode [3:0] (in _TIM3_CCMR1)
- #define _TIM3_IC1F0 ((uint8_t) (0x01 << 4))

  TIM3 Input capture 1 filter [0] (in _TIM3_CCMR1)
- #define _TIM3_IC1F1 ((uint8_t) (0x01 << 5))

  TIM3 Input capture 1 filter [1] (in _TIM3_CCMR1)
- #define _TIM3_IC1F2 ((uint8_t) (0x01 << 6))

  TIM3 Input capture 1 filter [2] (in _TIM3_CCMR1)
- #define _TIM3_IC1F3 ((uint8_t) (0x01 << 7))

  TIM3 Input capture 1 filter [3] (in _TIM3_CCMR1)
- #define _TIM3_CC2S ((uint8_t) (0x03 << 0))

  TIM3 Compare 2 selection [1:0] (in _TIM3_CCMR2)
- #define _TIM3_CC2S0 ((uint8_t) (0x01 << 0))

  TIM3 Compare 2 selection [0] (in _TIM3_CCMR2)
- #define _TIM3_CC2S1 ((uint8_t) (0x01 << 1))

  TIM3 Compare 2 selection [1] (in _TIM3_CCMR2)
- #define _TIM3_OC2FE ((uint8_t) (0x01 << 2))

  TIM3 Output compare 2 fast enable [0] (in _TIM3_CCMR2)
- #define _TIM3_OC2PE ((uint8_t) (0x01 << 3))

  TIM3 Output compare 2 preload enable [0] (in _TIM3_CCMR2)
- #define _TIM3_OC2M ((uint8_t) (0x07 << 4))

  TIM3 Output compare 2 mode [2:0] (in _TIM3_CCMR2)
- #define _TIM3_OC2M0 ((uint8_t) (0x01 << 4))

  TIM3 Output compare 2 mode [0] (in _TIM3_CCMR2)
- #define _TIM3_OC2M1 ((uint8_t) (0x01 << 5))

  TIM3 Output compare 2 mode [1] (in _TIM3_CCMR2)
- #define _TIM3_OC2M2 ((uint8_t) (0x01 << 6))

  TIM3 Output compare 2 mode [2] (in _TIM3_CCMR2)
- #define _TIM3_IC2PSC ((uint8_t) (0x03 << 2))

  TIM3 Input capture 2 prescaler [1:0] (in _TIM3_CCMR2)

- #define _TIM3_IC2PSC0 ((uint8_t) (0x01 << 2))
  TIM3 Input capture 2 prescaler [0] (in _TIM3_CCMR2)
- #define _TIM3_IC2PSC1 ((uint8_t) (0x01 << 3))
  TIM3 Input capture 2 prescaler [1] (in _TIM3_CCMR2)
- #define _TIM3_IC2F ((uint8_t) (0x0F << 4))
  TIM3 Output compare 2 mode [3:0] (in _TIM3_CCMR2)
- #define _TIM3_IC2F0 ((uint8_t) (0x01 << 4))
  TIM3 Input capture 2 filter [0] (in _TIM3_CCMR2)
- #define _TIM3_IC2F1 ((uint8_t) (0x01 << 5))
  TIM3 Input capture 2 filter [1] (in _TIM3_CCMR2)
- #define _TIM3_IC2F2 ((uint8_t) (0x01 << 6))
  TIM3 Input capture 2 filter [2] (in _TIM3_CCMR2)
- #define _TIM3_IC2F3 ((uint8_t) (0x01 << 7))
  TIM3 Input capture 2 filter [3] (in _TIM3_CCMR2)
- #define _TIM3_CC1E ((uint8_t) (0x01 << 0))
  TIM3 Capture/compare 1 output enable [0] (in _TIM3_CCER1)
- #define _TIM3_CC1P ((uint8_t) (0x01 << 1))
  TIM3 Capture/compare 1 output polarity [0] (in _TIM3_CCER1)
- #define _TIM3_CC2E ((uint8_t) (0x01 << 4))
  TIM3 Capture/compare 2 output enable [0] (in _TIM3_CCER1)
- #define _TIM3_CC2P ((uint8_t) (0x01 << 5))
  TIM3 Capture/compare 2 output polarity [0] (in _TIM3_CCER1)
- #define _TIM3_PSC ((uint8_t) (0x07 << 0))
  TIM3 prescaler [2:0] (in _TIM3_PSCR)
- #define _TIM3_PSC0 ((uint8_t) (0x01 << 0))
  TIM3 prescaler [0] (in _TIM3_PSCR)
- #define _TIM3_PSC1 ((uint8_t) (0x01 << 1))
  TIM3 prescaler [1] (in _TIM3_PSCR)
- #define _TIM3_PSC2 ((uint8_t) (0x01 << 2))
  TIM3 prescaler [2] (in _TIM3_PSCR)
- #define _TIM3_LOCK ((int8_t) (0x03 << 0))
  TIM3 Lock configuration [1:0] (in _TIM3_BKR)
- #define _TIM3_LOCK0 ((uint8_t) (0x01 << 0))
  TIM3 Lock configuration [0] (in _TIM3_BKR)
- #define _TIM3_LOCK1 ((uint8_t) (0x01 << 1))
  TIM3 Lock configuration [1] (in _TIM3_BKR)
- #define _TIM3_OSSI ((uint8_t) (0x01 << 2))
  TIM3 Off state selection for idle mode [0] (in _TIM3_BKR)
- #define _TIM3_BKE ((uint8_t) (0x01 << 4))
  TIM3 Break enable [0] (in _TIM3_BKR)
- #define _TIM3_BKP ((uint8_t) (0x01 << 5))
  TIM3 Break polarity [0] (in _TIM3_BKR)
- #define _TIM3_AOE ((uint8_t) (0x01 << 6))
  TIM3 Automatic output enable [0] (in _TIM3_BKR)
- #define _TIM3_MOE ((uint8_t) (0x01 << 7))
  TIM3 Main output enable [0] (in _TIM3_BKR)
- #define _TIM3_OIS1 ((uint8_t) (0x01 << 0))
  TIM3 Output idle state 1 (OC1 output) [0] (in _TIM3_OISR)
- #define _TIM3_OIS2 ((uint8_t) (0x01 << 2))
  TIM3 Output idle state 2 (OC2 output) [0] (in _TIM3_OISR)
- #define _TIM4 _SFR(TIM4_t, TIM4_AddressBase)

TIM4 struct/bit access.

- #define _TIM4_CR1 _SFR(uint8_t, TIM4_AddressBase+0x00)

  TIM4 control register 1.
- #define _TIM4_CR2 _SFR(uint8_t, TIM4_AddressBase+0x01)

  TIM4 control register 2.
- #define _TIM4_SMCR _SFR(uint8_t, TIM4_AddressBase+0x02)

  TIM4 Slave mode control register.
- #define _TIM4_IER _SFR(uint8_t, TIM4_AddressBase+0x03)

  TIM4 interrupt enable register.
- #define _TIM4_SR1 _SFR(uint8_t, TIM4_AddressBase+0x04)

  TIM4 status register.
- #define _TIM4_EGR _SFR(uint8_t, TIM4_AddressBase+0x05)

  TIM4 event generation register.
- #define _TIM4_CNTR _SFR(uint8_t, TIM4_AddressBase+0x06)

  TIM4 counter register.
- #define _TIM4_PSCR _SFR(uint8_t, TIM4_AddressBase+0x07)

  TIM4 clock prescaler register.
- #define _TIM4_ARR _SFR(uint8_t, TIM4_AddressBase+0x08)

  TIM4 auto-reload register.
- #define _TIM4_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM4 control register 1 reset value.
- #define _TIM4_CR2_RESET_VALUE ((uint8_t) 0x00)

  TIM4 control register 2 reset value.
- #define _TIM4_SMCR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 Slave mode control register reset value.
- #define _TIM4_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM4 interrupt enable register reset value.
- #define _TIM4_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM4 status register reset value.
- #define _TIM4_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 event generation register reset value.
- #define _TIM4_CNTR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 counter register reset value.
- #define _TIM4_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 clock prescaler register reset value.
- #define _TIM4_ARR_RESET_VALUE ((uint8_t) 0xFF)

  TIM4 auto-reload register reset value.
- #define _TIM4_CEN ((uint8_t) (0x01 << 0))

  TIM4 Counter enable [0] (in _TIM4_CR1)
- #define _TIM4_UDIS ((uint8_t) (0x01 << 1))

  TIM4 Update disable [0] (in _TIM4_CR1)
- #define _TIM4_URS ((uint8_t) (0x01 << 2))

  TIM4 Update request source [0] (in _TIM4_CR1)
- #define _TIM4_OPM ((uint8_t) (0x01 << 3))

  TIM4 One-pulse mode [0] (in _TIM4_CR1)
- #define _TIM4_ARPE ((uint8_t) (0x01 << 7))

  TIM4 Auto-reload preload enable [0] (in _TIM4_CR)
- #define _TIM4_MMS ((uint8_t) (0x07 << 4))

  TIM4 Master mode selection [2:0] (in _TIM4_CR2)
- #define _TIM4_MMS0 ((uint8_t) (0x01 << 4))

  TIM4 Master mode selection [0] (in _TIM4_CR2)

- #define _TIM4_MMS1 ((uint8_t) (0x01 << 5))

  TIM4 Master mode selection [1] (in _TIM4_CR2)
- #define _TIM4_MMS2 ((uint8_t) (0x01 << 6))

  TIM4 Master mode selection [2] (in _TIM4_CR2)
- #define _TIM4_SMS ((uint8_t) (0x07 << 0))

  TIM4 Clock/trigger/slave mode selection [2:0] (in _TIM4_SMCR)
- #define _TIM4_SMS0 ((uint8_t) (0x01 << 0))

  TIM4 Clock/trigger/slave mode selection [0] (in _TIM4_SMCR)
- #define _TIM4_SMS1 ((uint8_t) (0x01 << 1))

  TIM4 Clock/trigger/slave mode selection [1] (in _TIM4_SMCR)
- #define _TIM4_SMS2 ((uint8_t) (0x01 << 2))

  TIM4 Clock/trigger/slave mode selection [2] (in _TIM4_SMCR)
- #define _TIM4_TS ((uint8_t) (0x07 << 4))

  TIM4 Trigger selection [2:0] (in _TIM4_SMCR)
- #define _TIM4_TS0 ((uint8_t) (0x01 << 4))

  TIM4 Trigger selection [0] (in _TIM4_SMCR)
- #define _TIM4_TS1 ((uint8_t) (0x01 << 5))

  TIM4 Trigger selection [1] (in _TIM4_SMCR)
- #define _TIM4_TS2 ((uint8_t) (0x01 << 6))

  TIM4 Trigger selection [2] (in _TIM4_SMCR)
- #define _TIM4_MSM ((uint8_t) (0x01 << 7))

  TIM4 Master/slave mode [0] (in _TIM4_SMCR)
- #define _TIM4_UIE ((uint8_t) (0x01 << 0))

  TIM4 Update interrupt enable [0] (in _TIM4_IER)
- #define _TIM4_TIE ((uint8_t) (0x01 << 6))

  TIM4 Trigger interrupt enable [0] (in _TIM4_IER)
- #define _TIM4_UIF ((uint8_t) (0x01 << 0))

  TIM4 Update interrupt flag [0] (in _TIM4_SR1)
- #define _TIM4_TIF ((uint8_t) (0x01 << 6))

  TIM4 Trigger interrupt flag [0] (in _TIM4_SR1)
- #define _TIM4_UG ((uint8_t) (0x01 << 0))

  TIM4 Update generation [0] (in _TIM4_EGR)
- #define _TIM4_TG ((uint8_t) (0x01 << 6))

  TIM4 Trigger generation [0] (in _TIM4_EGR)
- #define _TIM4_PSC ((uint8_t) (0x0F << 0))

  TIM4 clock prescaler [3:0] (in _TIM4_PSCR)
- #define _TIM4_PSC0 ((uint8_t) (0x01 << 0))

  TIM4 clock prescaler [0] (in _TIM4_PSCR)
- #define _TIM4_PSC1 ((uint8_t) (0x01 << 1))

  TIM4 clock prescaler [1] (in _TIM4_PSCR)
- #define _TIM4_PSC2 ((uint8_t) (0x01 << 2))

  TIM4 clock prescaler [2] (in _TIM4_PSCR)
- #define _TIM4_PSC3 ((uint8_t) (0x01 << 3))

  TIM4 clock prescaler [3] (in _TIM4_PSCR)
- #define _IRTIM _SFR(IRTIM_t, IRTIM_AddressBase)

  IRTIM struct/bit access.
- #define _IRTIM_CR1 _SFR(uint8_t, IRTIM_AddressBase+0x00)

  IRTIM control register.
- #define _IRTIM_CR_RESET_VALUE ((uint8_t) 0x00)

  IRTIM control register reset value.
- #define _IRTIM_IR_EN ((uint8_t) (0x01 << 0))

IRTIM Infrared output enable [0] (in _IRTIM_CR)

- #define _IRTIM_HS_EN ((uint8_t) (0x01 << 1))

  IRTIM High Sink LED driver enable [0] (in _IRTIM_CR)
- #define _COMP _SFR(COMP_t_t, COMP_AddressBase)

  COMP struct/bit access.
- #define _COMP_CR _SFR(uint8_t, COMP_AddressBase+0x00)

  Comparator control register.
- #define _COMP_CSR _SFR(uint8_t, COMP_AddressBase+0x01)

  Comparator control status register.
- #define _COMP_CCS _SFR(uint8_t, COMP_AddressBase+0x02)

  Comparator channel selection.
- #define _COMP_CR_RESET_VALUE ((uint8_t) 0x00)

  Comparator control register reset value.
- #define _COMP_CSR_RESET_VALUE ((uint8_t) 0x00)

  Comparator control status register reset value.
- #define _COMP_CCS_RESET_VALUE ((uint8_t) 0x00)

  Comparator channel selection reset value.
- #define _COMP_BIAS_EN ((uint8_t) (0x01 << 0))

  COMP Bias enable [0] (in _COMP_CR)
- #define _COMP_COMP1_EN ((uint8_t) (0x01 << 1))

  COMP First comparator enable [0] (in _COMP_CR)
- #define _COMP_COMP2_EN ((uint8_t) (0x01 << 2))

  COMP Second comparator enable [0] (in _COMP_CR)
- #define _COMP_COMPREF ((uint8_t) (0x01 << 3))

  COMP Comparator reference [0] (in _COMP_CR)
- #define _COMP_POL ((uint8_t) (0x01 << 4))

  COMP Comparator polarity [0] (in _COMP_CR)
- #define _COMP_CNF_TIM ((uint8_t) (0x03 << 5))

  COMP Comparator 1/2 output connected to TIM2/3 capture or break [1:0] (in _COMP_CR)
- #define _COMP_CNF_TIM0 ((uint8_t) (0x03 << 5))

  COMP Comparator 1/2 output connected to TIM2/3 capture or break [0] (in _COMP_CR)
- #define _COMP_CNF_TIM1 ((uint8_t) (0x03 << 6))

  COMP Comparator 1/2 output connected to TIM2/3 capture or break [1] (in _COMP_CR)
- #define _COMP_IC1_BK ((uint8_t) (0x01 << 7))

  COMP Input capture 1 / break selection [0] (in _COMP_CR)
- #define _COMP_COMP1_OUT ((uint8_t) (0x01 << 0))

  COMP First comparator output [0] (in _COMP_CSR)
- #define _COMP_COMP2_OUT ((uint8_t) (0x01 << 1))

  COMP Second comparator output [0] (in _COMP_CSR)
- #define _COMP_CEF1 ((uint8_t) (0x01 << 4))

  COMP First comparator event flag [0] (in _COMP_CSR)
- #define _COMP_ITEN1 ((uint8_t) (0x01 << 5))

  COMP First comparator interrupt enable [0] (in _COMP_CSR)
- #define _COMP_CEF2 ((uint8_t) (0x01 << 6))

  COMP Second comparator event flag [0] (in _COMP_CSR)
- #define _COMP_ITEN2 ((uint8_t) (0x01 << 7))

  COMP Second comparator interrupt enable [0] (in _COMP_CSR)
- #define _COMP_COMP1_CH1 ((uint8_t) (0x01 << 0))

  COMP Comparator 1 switch 1 enable [0] (in _COMP_CCS)
- #define _COMP_COMP1_CH2 ((uint8_t) (0x01 << 1))

  COMP Comparator 1 switch 2 enable [0] (in _COMP_CCS)

- #define _COMP_COMP1_CH3 ((uint8_t) (0x01 << 2))
  COMP Comparator 1 switch 3 enable [0] (in _COMP_CCS)
- #define _COMP_COMP1_CH4 ((uint8_t) (0x01 << 3))
  COMP Comparator 1 switch 4 enable [0] (in _COMP_CCS)
- #define _COMP_COMP2_CH1 ((uint8_t) (0x01 << 4))
  COMP Comparator 2 switch 1 enable [0] (in _COMP_CCS)
- #define _COMP_COMP2_CH2 ((uint8_t) (0x01 << 5))
  COMP Comparator 2 switch 2 enable [0] (in _COMP_CCS)
- #define _COMP_COMP2_CH3 ((uint8_t) (0x01 << 6))
  COMP Comparator 2 switch 3 enable [0] (in _COMP_CCS)
- #define _COMP_COMP2_CH4 ((uint8_t) (0x01 << 7))
  COMP Comparator 2 switch 4 enable [0] (in _COMP_CCS)
- #define _CFG _SFR(CFG_t, CFG_AddressBase)
  CFG struct/bit access.
- #define _CFG_GCR _SFR(uint8_t, CFG_AddressBase+0x00)
  Global configuration register (CFG_GCR)
- #define _CFG_GCR_RESET_VALUE ((uint8_t)0x00)
- #define _CFG_SWD ((uint8_t) (0x01 << 0))
  SWIM disable [0].
- #define _CFG_AL ((uint8_t) (0x01 << 1))
  Activation level [0].
- #define _ITC _SFR(ITC_t, ITC_AddressBase)
  ITC struct/bit access.
- #define _ITC_SPR1 _SFR(uint8_t, ITC_AddressBase+0x00)
  Interrupt priority register 1/8.
- #define _ITC_SPR2 _SFR(uint8_t, ITC_AddressBase+0x01)
  Interrupt priority register 2/8.
- #define _ITC_SPR3 _SFR(uint8_t, ITC_AddressBase+0x02)
  Interrupt priority register 3/8.
- #define _ITC_SPR4 _SFR(uint8_t, ITC_AddressBase+0x03)
  Interrupt priority register 4/8.
- #define _ITC_SPR5 _SFR(uint8_t, ITC_AddressBase+0x04)
  Interrupt priority register 5/8.
- #define _ITC_SPR6 _SFR(uint8_t, ITC_AddressBase+0x05)
  Interrupt priority register 6/8.
- #define _ITC_SPR7 _SFR(uint8_t, ITC_AddressBase+0x06)
  Interrupt priority register 7/8.
- #define _ITC_SPR8 _SFR(uint8_t, ITC_AddressBase+0x07)
  Interrupt priority register 8/8.
- #define _ITC_SPR1_RESET_VALUE ((uint8_t) 0xFF)
  Interrupt priority register 1/8 reset value.
- #define _ITC_SPR2_RESET_VALUE ((uint8_t) 0xFF)
  Interrupt priority register 2/8 reset value.
- #define _ITC_SPR3_RESET_VALUE ((uint8_t) 0xFF)
  Interrupt priority register 3/8 reset value.
- #define _ITC_SPR4_RESET_VALUE ((uint8_t) 0xFF)
  Interrupt priority register 4/8 reset value.
- #define _ITC_SPR5_RESET_VALUE ((uint8_t) 0xFF)
  Interrupt priority register 5/8 reset value.
- #define _ITC_SPR6_RESET_VALUE ((uint8_t) 0xFF)
  Interrupt priority register 6/8 reset value.

- #define _ITC_SPR7_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 7/8 reset value.
- #define _ITC_SPR8_RESET_VALUE ((uint8_t) 0x0F)

  Interrupt priority register 8/8 reset value.
- #define _ITC_VECT1SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 1 [1:0] (in _ITC_SPR1)
- #define _ITC_VECT1SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 1 [0] (in _ITC_SPR1)
- #define _ITC_VECT1SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 1 [1] (in _ITC_SPR1)
- #define _ITC_VECT4SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 4 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT4SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 4 [0] (in _ITC_SPR2)
- #define _ITC_VECT4SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 4 [1] (in _ITC_SPR2)
- #define _ITC_VECT6SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 6 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT6SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 6 [0] (in _ITC_SPR2)
- #define _ITC_VECT6SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 6 [1] (in _ITC_SPR2)
- #define _ITC_VECT7SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 7 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT7SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 7 [0] (in _ITC_SPR2)
- #define _ITC_VECT7SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 7 [1] (in _ITC_SPR2)
- #define _ITC_VECT8SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 8 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT8SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 8 [0] (in _ITC_SPR3)
- #define _ITC_VECT8SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 8 [1] (in _ITC_SPR3)
- #define _ITC_VECT9SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 9 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT9SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 9 [0] (in _ITC_SPR3)
- #define _ITC_VECT9SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 9 [1] (in _ITC_SPR3)
- #define _ITC_VECT10SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 10 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT10SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 10 [0] (in _ITC_SPR3)
- #define _ITC_VECT10SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 10 [1] (in _ITC_SPR3)
- #define _ITC_VECT11SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 11 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT11SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 11 [0] (in _ITC_SPR3)
- #define _ITC_VECT11SPR1 ((uint8_t) (0x01 << 7))

ITC interrupt priority vector 11 [1] (in _ITC_SPR3)

- #define _ITC_VECT12SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 12 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT12SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 12 [0] (in _ITC_SPR4)
- #define _ITC_VECT12SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 12 [1] (in _ITC_SPR4)
- #define _ITC_VECT13SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 13 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT13SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 13 [0] (in _ITC_SPR4)
- #define _ITC_VECT13SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 13 [1] (in _ITC_SPR4)
- #define _ITC_VECT14SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 14 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT14SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 14 [0] (in _ITC_SPR4)
- #define _ITC_VECT14SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 14 [1] (in _ITC_SPR4)
- #define _ITC_VECT15SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 15 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT15SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 15 [0] (in _ITC_SPR4)
- #define _ITC_VECT15SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 15 [1] (in _ITC_SPR4)
- #define _ITC_VECT19SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 19 [1:0] (in _ITC_SPR5)
- #define _ITC_VECT19SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 19 [0] (in _ITC_SPR5)
- #define _ITC_VECT19SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 19 [1] (in _ITC_SPR5)
- #define _ITC_VECT20SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 20 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT20SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 20 [0] (in _ITC_SPR6)
- #define _ITC_VECT20SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 20 [1] (in _ITC_SPR6)
- #define _ITC_VECT21SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 21 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT21SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 21 [0] (in _ITC_SPR6)
- #define _ITC_VECT21SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 21 [1] (in _ITC_SPR6)
- #define _ITC_VECT22SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 22 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT22SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 22 [0] (in _ITC_SPR6)
- #define _ITC_VECT22SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 22 [1] (in _ITC_SPR6)
- #define _ITC_VECT25SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 25 [1:0] (in _ITC_SPR7)

- #define _ITC_VECT25SPR0 ((uint8_t) (0x01 << 2))
  ITC interrupt priority vector 25 [0] (in _ITC_SPR7)
- #define _ITC_VECT25SPR1 ((uint8_t) (0x01 << 3))
  ITC interrupt priority vector 25 [1] (in _ITC_SPR7)
- #define _ITC_VECT26SPR ((uint8_t) (0x03 << 4))
  ITC interrupt priority vector 26 [1:0] (in _ITC_SPR7)
- #define _ITC_VECT26SPR0 ((uint8_t) (0x01 << 4))
  ITC interrupt priority vector 26 [0] (in _ITC_SPR7)
- #define _ITC_VECT26SPR1 ((uint8_t) (0x01 << 5))
  ITC interrupt priority vector 26 [1] (in _ITC_SPR7)
- #define _ITC_VECT27SPR ((uint8_t) (0x03 << 6))
  ITC interrupt priority vector 27 [1:0] (in _ITC_SPR7)
- #define _ITC_VECT27SPR0 ((uint8_t) (0x01 << 6))
  ITC interrupt priority vector 27 [0] (in _ITC_SPR7)
- #define _ITC_VECT27SPR1 ((uint8_t) (0x01 << 7))
  ITC interrupt priority vector 27 [1] (in _ITC_SPR7)
- #define _ITC_VECT28SPR ((uint8_t) (0x03 << 0))
  ITC interrupt priority vector 28 [1:0] (in _ITC_SPR8)
- #define _ITC_VECT28SPR0 ((uint8_t) (0x01 << 0))
  ITC interrupt priority vector 28 [0] (in _ITC_SPR8)
- #define _ITC_VECT28SPR1 ((uint8_t) (0x01 << 1))
  ITC interrupt priority vector 28 [1] (in _ITC_SPR8)
- #define _ITC_VECT29SPR ((uint8_t) (0x03 << 2))
  ITC interrupt priority vector 29 [1:0] (in _ITC_SPR8)
- #define _ITC_VECT29SPR0 ((uint8_t) (0x01 << 2))
  ITC interrupt priority vector 29 [0] (in _ITC_SPR8)
- #define _ITC_VECT29SPR1 ((uint8_t) (0x01 << 3))
  ITC interrupt priority vector 29 [1] (in _ITC_SPR8)

5.2.1 Detailed Description

5.2.2 Macro Definition Documentation

5.2.2.1 __AWU_VECTOR__

```
#define __AWU_VECTOR__ 4
```

irq4 - Auto Wake Up from Halt interrupt (AWU)

Definition at line 246 of file STM8L10x.h.

5.2.2.2 __COMP_VECTOR__

```
#define __COMP_VECTOR__ 18
```

irq18 - comparator interrupt

Definition at line 260 of file STM8L10x.h.

5.2.2.3 __EXTI0_VECTOR__

```
#define __EXTI0_VECTOR__ 8
```

irq8 - External interrupt 0

Definition at line 250 of file STM8L10x.h.

5.2.2.4 __EXTI1_VECTOR__

```
#define __EXTI1_VECTOR__ 9
```

irq9 - External interrupt 1

Definition at line 251 of file STM8L10x.h.

5.2.2.5 __EXTI2_VECTOR__

```
#define __EXTI2_VECTOR__ 10
```

irq10 - External interrupt 2

Definition at line 252 of file STM8L10x.h.

5.2.2.6 __EXTI3_VECTOR__

```
#define __EXTI3_VECTOR__ 11
```

irq11 - External interrupt 3

Definition at line 253 of file STM8L10x.h.

5.2.2.7 __EXTI4_VECTOR__

```
#define __EXTI4_VECTOR__ 12
```

irq12 - External interrupt 4

Definition at line 254 of file STM8L10x.h.

5.2.2.8 __EXTI5_VECTOR__

```
#define __EXTI5_VECTOR__ 13
```

irq13 - External interrupt 5

Definition at line 255 of file STM8L10x.h.

5.2.2.9 __EXTI6_VECTOR__

```
#define __EXTI6_VECTOR__ 14
```

irq14 - External interrupt 6

Definition at line 256 of file STM8L10x.h.

5.2.2.10 __EXTI7_VECTOR__

```
#define __EXTI7_VECTOR__ 15
```

irq15 - External interrupt 7

Definition at line 257 of file STM8L10x.h.

5.2.2.11 __FLASH_VECTOR__

```
#define __FLASH_VECTOR__ 1
```

irq1 - flash interrupt

Definition at line 243 of file STM8L10x.h.

5.2.2.12 __I2C_VECTOR__

```
#define __I2C_VECTOR__ 19
```

irq29 - I2C interrupt

Definition at line 271 of file STM8L10x.h.

5.2.2.13 __PORTB_VECTOR__

```
#define __PORTB_VECTOR__ 6
```

irq6 - External interrupt port B

Definition at line 248 of file STM8L10x.h.

5.2.2.14 __PORTD_VECTOR__

```
#define __PORTD_VECTOR__ 7
```

irq7 - External interrupt port D

Definition at line 249 of file STM8L10x.h.

5.2.2.15 __SPI_VECTOR__

```
#define __SPI_VECTOR__ 26
```

irq26 - SPI End of transfer interrupt

Definition at line 268 of file STM8L10x.h.

5.2.2.16 __TIM2_CAPCOM_VECTOR__

```
#define __TIM2_CAPCOM_VECTOR__ 20
```

irq20 - TIM2 Capture/Compare interrupt

Definition at line 262 of file STM8L10x.h.

5.2.2.17 __TIM2_UPD_OVF_VECTOR__

```
#define __TIM2_UPD_OVF_VECTOR__ 19
```

irq19 - TIM2 Update/overflow/trigger/break interrupt

Definition at line 261 of file STM8L10x.h.

5.2.2.18 __TIM3_CAPCOM_VECTOR__

```
#define __TIM3_CAPCOM_VECTOR__ 22
```

irq22 - TIM3 Capture/Compare interrupt

Definition at line 264 of file STM8L10x.h.

5.2.2.19 __TIM3_UPD_OVF_VECTOR__

```
#define __TIM3_UPD_OVF_VECTOR__ 21
```

irq21 - TIM3 Update/overflow/break interrupt

Definition at line 263 of file STM8L10x.h.

5.2.2.20 __TIM4_UPD_VECTOR__

```
#define __TIM4_UPD_VECTOR__ 25
```

irq25 - TIM4 Update/trigger interrupt

Definition at line 267 of file STM8L10x.h.

5.2.2.21 __USART_RXF_VECTOR__

```
#define __USART_RXF_VECTOR__ 28
```

irq28 - USART receive (RX full) interrupt

Definition at line 270 of file STM8L10x.h.

5.2.2.22 __USART_TXE_VECTOR__

```
#define __USART_TXE_VECTOR__ 27
```

irq27 - USART send (TX empty) interrupt

Definition at line 269 of file STM8L10x.h.

5.2.2.23 _AWU

```
#define _AWU _SFR(AWU_t, AWU_AddressBase)
```

Auto Wake-Up struct/bit access.

Definition at line 867 of file STM8L10x.h.

5.2.2.24 _AWU_APR

```
#define _AWU_APR _SFR(uint8_t, AWU_AddressBase+0x01)
```

Auto Wake-Up Asynchronous prescaler register (AWU_APR)

Definition at line 869 of file STM8L10x.h.

5.2.2.25 _AWU_APR_RESET_VALUE

```
#define _AWU_APR_RESET_VALUE ((uint8_t) 0x3F)
```

Auto Wake-Up Asynchronous prescaler register reset value.

Definition at line 874 of file STM8L10x.h.

5.2.2.26 _AWU_APRE

```
#define _AWU_APRE ((uint8_t) (0x3F << 0))
```

Auto-wakeup asynchronous prescaler divider [5:0] (in _AWU_APR)

Definition at line 885 of file STM8L10x.h.

5.2.2.27 _AWU_APRE0

```
#define _AWU_APRE0 ((uint8_t) (0x01 << 0))
```

Auto-wakeup asynchronous prescaler divider [0] (in _AWU_APR)

Definition at line 886 of file STM8L10x.h.

5.2.2.28 _AWU_APRE1

```
#define _AWU_APRE1 ((uint8_t) (0x01 << 1))
```

Auto-wakeup asynchronous prescaler divider [1] (in _AWU_APR)

Definition at line 887 of file STM8L10x.h.

5.2.2.29 _AWU_APRE2

```
#define _AWU_APRE2 ((uint8_t) (0x01 << 2))
```

Auto-wakeup asynchronous prescaler divider [2] (in _AWU_APR)

Definition at line 888 of file STM8L10x.h.

5.2.2.30 _AWU_APRE3

```
#define _AWU_APRE3 ((uint8_t) (0x01 << 3))
```

Auto-wakeup asynchronous prescaler divider [3] (in _AWU_APR)

Definition at line 889 of file STM8L10x.h.

5.2.2.31 _AWU_APRE4

```
#define _AWU_APRE4 ((uint8_t) (0x01 << 4))
```

Auto-wakeup asynchronous prescaler divider [4] (in _AWU_APR)

Definition at line 890 of file STM8L10x.h.

5.2.2.32 _AWU_APRE5

```
#define _AWU_APRE5 ((uint8_t) (0x01 << 5))
```

Auto-wakeup asynchronous prescaler divider [5] (in _AWU_APR)

Definition at line 891 of file STM8L10x.h.

5.2.2.33 _AWU_AWUEN

```
#define _AWU_AWUEN ((uint8_t) (0x01 << 4))
```

Auto-wakeup enable [0] (in _AWU_CSR)

Definition at line 880 of file STM8L10x.h.

5.2.2.34 _AWU_AWUF

```
#define _AWU_AWUF ((uint8_t) (0x01 << 5))
```

Auto-wakeup status flag [0] (in _AWU_CSR)

Definition at line 881 of file STM8L10x.h.

5.2.2.35 _AWU_AWUTB

```
#define _AWU_AWUTB ((uint8_t) (0x0F << 0))
```

Auto-wakeup timebase selection [3:0] (in _AWU_APR)

Definition at line 895 of file STM8L10x.h.

5.2.2.36 _AWU_AWUTB0

```
#define _AWU_AWUTB0 ((uint8_t) (0x01 << 0))
```

Auto-wakeup timebase selection [0] (in _AWU_APR)

Definition at line 896 of file STM8L10x.h.

5.2.2.37 _AWU_AWUTB1

```
#define _AWU_AWUTB1 ((uint8_t) (0x01 << 1))
```

Auto-wakeup timebase selection [1] (in _AWU_APR)

Definition at line 897 of file STM8L10x.h.

5.2.2.38 _AWU_AWUTB2

```
#define _AWU_AWUTB2 ((uint8_t) (0x01 << 2))
```

Auto-wakeup timebase selection [2] (in _AWU_APR)

Definition at line 898 of file STM8L10x.h.

5.2.2.39 _AWU_AWUTB3

```
#define _AWU_AWUTB3 ((uint8_t) (0x01 << 3))
```

Auto-wakeup timebase selection [3] (in _AWU_APR)

Definition at line 899 of file STM8L10x.h.

5.2.2.40 _AWU_CSR

```
#define _AWU_CSR _SFR(uint8_t, AWU_AddressBase+0x00)
```

Auto Wake-Up Control/status register (AWU_CSR)

Definition at line 868 of file STM8L10x.h.

5.2.2.41 _AWU_CSR_RESET_VALUE

```
#define _AWU_CSR_RESET_VALUE ((uint8_t) 0x00)
```

Auto Wake-Up Control/status register reset value.

Definition at line 873 of file STM8L10x.h.

5.2.2.42 _AWU_MSR

```
#define _AWU_MSR ((uint8_t) (0x01 << 0))
```

Auto Wake-Up LSI measurement enable [0] (in _AWU_CSR)

Definition at line 878 of file STM8L10x.h.

5.2.2.43 _AWU_TBR

```
#define _AWU_TBR _SFR(uint8_t, AWU_AddressBase+0x02)
```

Auto Wake-Up Timebase selection register (AWU_TBR)

Definition at line 870 of file STM8L10x.h.

5.2.2.44 _AWU_TBR_RESET_VALUE

```
#define _AWU_TBR_RESET_VALUE ((uint8_t) 0x00)
```

Auto Wake-Up Timebase selection register reset value.

Definition at line 875 of file STM8L10x.h.

5.2.2.45 _BEEP

```
#define _BEEP _SFR(BEEP_t, BEEP_AddressBase)
```

Beeper struct/bit access.

Definition at line 924 of file STM8L10x.h.

5.2.2.46 _BEEP_BEEPDIV

```
#define _BEEP_BEEPDIV ((uint8_t) (0x1F << 0))
```

Beeper clock prescaler divider [4:0] (in _BEEP_CSR)

Definition at line 931 of file STM8L10x.h.

5.2.2.47 _BEEP_BEEPDIV0

```
#define _BEEP_BEEPDIV0 ((uint8_t) (0x01 << 0))
```

Beeper clock prescaler divider [0] (in _BEEP_CSR)

Definition at line 932 of file STM8L10x.h.

5.2.2.48 _BEEP_BEEPDIV1

```
#define _BEEP_BEEPDIV1 ((uint8_t) (0x01 << 1))
```

Beeper clock prescaler divider [1] (in _BEEP_CSR)

Definition at line 933 of file STM8L10x.h.

5.2.2.49 _BEEP_BEEPDIV2

```
#define _BEEP_BEEPDIV2 ((uint8_t) (0x01 << 2))
```

Beeper clock prescaler divider [2] (in _BEEP_CSR)

Definition at line 934 of file STM8L10x.h.

5.2.2.50 _BEEP_BEEPDIV3

```
#define _BEEP_BEEPDIV3 ((uint8_t) (0x01 << 3))
```

Beeper clock prescaler divider [3] (in _BEEP_CSR)

Definition at line 935 of file STM8L10x.h.

5.2.2.51 _BEEP_BEEPDIV4

```
#define _BEEP_BEEPDIV4 ((uint8_t) (0x01 << 4))
```

Beeper clock prescaler divider [4] (in _BEEP_CSR)

Definition at line 936 of file STM8L10x.h.

5.2.2.52 _BEEP_BEEPEN

```
#define _BEEP_BEEPEN ((uint8_t) (0x01 << 5))
```

Beeper enable [0] (in _BEEP_CSR)

Definition at line 937 of file STM8L10x.h.

5.2.2.53 _BEEP_BEEPSEL

```
#define _BEEP_BEEPSEL ((uint8_t) (0x03 << 6))
```

Beeper frequency selection [1:0] (in _BEEP_CSR)

Definition at line 938 of file STM8L10x.h.

5.2.2.54 _BEEP_BEEPSEL0

```
#define _BEEP_BEEPSEL0 ((uint8_t) (0x01 << 6))
```

Beeper frequency selection [0] (in _BEEP_CSR)

Definition at line 939 of file STM8L10x.h.

5.2.2.55 _BEEP_BEEPSEL1

```
#define _BEEP_BEEPSEL1 ((uint8_t) (0x01 << 7))
```

Beeper frequency selection [1] (in _BEEP_CSR)

Definition at line 940 of file STM8L10x.h.

5.2.2.56 _BEEP_CSR

```
#define _BEEP_CSR _SFR(uint8_t, BEEP_AddressBase+0x00)
```

Beeper control/status register (BEEP_CSR)

Definition at line 925 of file STM8L10x.h.

5.2.2.57 _BEEP_CSR_RESET_VALUE

```
#define _BEEP_CSR_RESET_VALUE ((uint8_t) 0x1F)
```

Beeper control/status register reset value.

Definition at line 928 of file STM8L10x.h.

5.2.2.58 _BITS

```
#define _BITS unsigned int
```

data type in bit structs (follow C90 standard)

Definition at line 177 of file STM8L10x.h.

5.2.2.59 _CFG

```
#define _CFG _SFR(CFG_t, CFG_AddressBase)
```

CFG struct/bit access.

Definition at line 2479 of file STM8L10x.h.

5.2.2.60 _CFG_AL

```
#define _CFG_AL ((uint8_t) (0x01 << 1))
```

Activation level [0].

Definition at line 2487 of file STM8L10x.h.

5.2.2.61 _CFG_GCR

```
#define _CFG_GCR _SFR(uint8_t, CFG_AddressBase+0x00)
```

Global configuration register (CFG_GCR)

Definition at line 2480 of file STM8L10x.h.

5.2.2.62 _CFG_GCR_RESET_VALUE

```
#define _CFG_GCR_RESET_VALUE ((uint8_t)0x00)
```

Definition at line 2483 of file STM8L10x.h.

5.2.2.63 _CFG_SWD

```
#define _CFG_SWD ((uint8_t) (0x01 << 0))
```

SWIM disable [0].

Definition at line 2486 of file STM8L10x.h.

5.2.2.64 _CLK

```
#define _CLK _SFR(CLK_t, CLK_AddressBase)
```

Clock module struct/bit access.

Definition at line 729 of file STM8L10x.h.

5.2.2.65 _CLK_AWU_BEEP

```
#define _CLK_AWU_BEEP ((uint8_t) (0x01 << 6))
```

clock enable AWU/BEEP [0] (in _CLK_PCKENR1)

Definition at line 759 of file STM8L10x.h.

5.2.2.66 _CLK_CCOEN

```
#define _CLK_CCOEN ((uint8_t) (0x01 << 0))
```

Configurable clock output enable [0] (in _CLK_CCOR)

Definition at line 763 of file STM8L10x.h.

5.2.2.67 _CLK_CCOR

```
#define _CLK_CCOR _SFR(uint8_t, CLK_AddressBase+0x05)
```

Configurable clock output register.

Definition at line 734 of file STM8L10x.h.

5.2.2.68 _CLK_CCOR_RESET_VALUE

```
#define _CLK_CCOR_RESET_VALUE ((uint8_t) 0x00)
```

Configurable clock output register reset value.

Definition at line 739 of file STM8L10x.h.

5.2.2.69 _CLK_CCOSEL

```
#define _CLK_CCOSEL ((uint8_t) (0x03 << 1))
```

Configurable clock output selection [1:0] (in _CLK_CCOR)

Definition at line 764 of file STM8L10x.h.

5.2.2.70 _CLK_CCOSEL0

```
#define _CLK_CCOSEL0 ((uint8_t) (0x01 << 1))
```

Configurable clock output selection [0] (in _CLK_CCOR)

Definition at line 765 of file STM8L10x.h.

5.2.2.71 _CLK_CCOSEL1

```
#define _CLK_CCOSEL1 ((uint8_t) (0x01 << 2))
```

Configurable clock output selection [1] (in _CLK_CCOR)

Definition at line 766 of file STM8L10x.h.

5.2.2.72 _CLK_CCOSEL_DIV1

```
#define _CLK_CCOSEL_DIV1 ((uint8_t) (0x00 << 1))
```

set clock output selection to 1 (in _CLK_CCOR)

Definition at line 769 of file STM8L10x.h.

5.2.2.73 _CLK_CCOSEL_DIV16

```
#define _CLK_CCOSEL_DIV16 ((uint8_t) (0x03 << 1))
```

set clock output selection to 1/16 (in _CLK_CCOR)

Definition at line 772 of file STM8L10x.h.

5.2.2.74 _CLK_CCOSEL_DIV2

```
#define _CLK_CCOSEL_DIV2 ((uint8_t) (0x01 << 1))
```

set clock output selection to 1/2 (in _CLK_CCOR)

Definition at line 770 of file STM8L10x.h.

5.2.2.75 _CLK_CCOSEL_DIV4

```
#define _CLK_CCOSEL_DIV4 ((uint8_t) (0x02 << 1))
```

set clock output selection to 1/4 (in _CLK_CCOR)

Definition at line 771 of file STM8L10x.h.

5.2.2.76 _CLK_CKDIVR

```
#define _CLK_CKDIVR _SFR(uint8_t, CLK_AddressBase+0x00)
```

Clock Divider Register.

Definition at line 730 of file STM8L10x.h.

5.2.2.77 _CLK_CKDIVR_RESET_VALUE

```
#define _CLK_CKDIVR_RESET_VALUE ((uint8_t) 0x03)
```

Clock divider register reset value.

Definition at line 737 of file STM8L10x.h.

5.2.2.78 _CLK_HSIDIV

```
#define _CLK_HSIDIV ((uint8_t) (0x03 << 0))
```

High speed internal clock prescaler [1:0] (in _CLK_CKDIVR)

Definition at line 742 of file STM8L10x.h.

5.2.2.79 _CLK_HSIDIV0

```
#define _CLK_HSIDIV0 ((uint8_t) (0x01 << 0))
```

High speed internal clock prescaler [0] (in _CLK_CKDIVR)

Definition at line 743 of file STM8L10x.h.

5.2.2.80 _CLK_HSIDIV1

```
#define _CLK_HSIDIV1 ((uint8_t) (0x01 << 1))
```

High speed internal clock prescaler [1] (in _CLK_CKDIVR)

Definition at line 744 of file STM8L10x.h.

5.2.2.81 _CLK_HSIDIV_DIV1

```
#define _CLK_HSIDIV_DIV1 ((uint8_t) (0x00 << 0))
```

set HSI prescaler to 1 (in _CLK_CKDIVR)

Definition at line 747 of file STM8L10x.h.

5.2.2.82 _CLK_HSIDIV_DIV2

```
#define _CLK_HSIDIV_DIV2 ((uint8_t) (0x01 << 0))
```

set HSI prescaler to 1/2 (in _CLK_CKDIVR)

Definition at line 748 of file STM8L10x.h.

5.2.2.83 _CLK_HSIDIV_DIV4

```
#define _CLK_HSIDIV_DIV4 ((uint8_t) (0x02 << 0))
```

set HSI prescaler to 1/4 (in _CLK_CKDIVR)

Definition at line 749 of file STM8L10x.h.

5.2.2.84 _CLK_HSIDIV_DIV8

```
#define _CLK_HSIDIV_DIV8 ((uint8_t) (0x03 << 0))
```

set HSI prescaler to 1/8 (in _CLK_CKDIVR)

Definition at line 750 of file STM8L10x.h.

5.2.2.85 _CLK_I2C

```
#define _CLK_I2C ((uint8_t) (0x01 << 3))
```

clock enable I2C [0] (in _CLK_PCKENR1)

Definition at line 756 of file STM8L10x.h.

5.2.2.86 _CLK_PCKENR

```
#define _CLK_PCKENR _SFR(uint8_t, CLK_AddressBase+0x03)
```

Peripheral clock gating register.

Definition at line 732 of file STM8L10x.h.

5.2.2.87 _CLK_PCKENR_RESET_VALUE

```
#define _CLK_PCKENR_RESET_VALUE ((uint8_t) 0x00)
```

Peripheral clock gating register reset value.

Definition at line 738 of file STM8L10x.h.

5.2.2.88 _CLK_SPI

```
#define _CLK_SPI ((uint8_t) (0x01 << 4))
```

clock enable SPI [0] (in _CLK_PCKENR1)

Definition at line 757 of file STM8L10x.h.

5.2.2.89 _CLK_TIM2

```
#define _CLK_TIM2 ((uint8_t) (0x01 << 0))
```

clock enable TIM2 [0] (in _CLK_PCKENR1)

Definition at line 753 of file STM8L10x.h.

5.2.2.90 _CLK_TIM3

```
#define _CLK_TIM3 ((uint8_t) (0x01 << 1))
```

clock enable TIM3 [0] (in _CLK_PCKENR1)

Definition at line 754 of file STM8L10x.h.

5.2.2.91 _CLK_TIM4

```
#define _CLK_TIM4 ((uint8_t) (0x01 << 2))
```

clock enable TIM4 [0] (in _CLK_PCKENR1)

Definition at line 755 of file STM8L10x.h.

5.2.2.92 _CLK_USART

```
#define _CLK_USART ((uint8_t) (0x01 << 5))
```

clock enable USART [0] (in _CLK_PCKENR1)

Definition at line 758 of file STM8L10x.h.

5.2.2.93 _COMP

```
#define _COMP _SFR(COMP_t_t, COMP_AddressBase)
```

COMP struct/bit access.

Definition at line 2417 of file STM8L10x.h.

5.2.2.94 _COMP_BIAS_EN

```
#define _COMP_BIAS_EN ((uint8_t) (0x01 << 0))
```

COMP Bias enable [0] (in _COMP_CR)

Definition at line 2428 of file STM8L10x.h.

5.2.2.95 _COMP_CCS

```
#define _COMP_CCS _SFR(uint8_t, COMP_AddressBase+0x02)
```

Comparator channel selection.

Definition at line 2420 of file STM8L10x.h.

5.2.2.96 _COMP_CCS_RESET_VALUE

```
#define _COMP_CCS_RESET_VALUE ((uint8_t) 0x00)
```

Comparator channel selection reset value.

Definition at line 2425 of file STM8L10x.h.

5.2.2.97 _COMP_CEF1

```
#define _COMP_CEF1 ((uint8_t) (0x01 << 4))
```

COMP First comparator event flag [0] (in _COMP_CSR)

Definition at line 2442 of file STM8L10x.h.

5.2.2.98 _COMP_CEF2

```
#define _COMP_CEF2 ((uint8_t) (0x01 << 6))
```

COMP Second comparator event flag [0] (in _COMP_CSR)

Definition at line 2444 of file STM8L10x.h.

5.2.2.99 _COMP_CNF_TIM

```
#define _COMP_CNF_TIM ((uint8_t) (0x03 << 5))
```

COMP Comparator 1/2 output connected to TIM2/3 capture or break [1:0] (in _COMP_CR)

Definition at line 2433 of file STM8L10x.h.

5.2.2.100 _COMP_CNF_TIM0

```
#define _COMP_CNF_TIM0 ((uint8_t) (0x03 << 5))
```

COMP Comparator 1/2 output connected to TIM2/3 capture or break [0] (in _COMP_CR)

Definition at line 2434 of file STM8L10x.h.

5.2.2.101 _COMP_CNF_TIM1

```
#define _COMP_CNF_TIM1 ((uint8_t) (0x03 << 6))
```

COMP Comparator 1/2 output connected to TIM2/3 capture or break [1] (in _COMP_CR)

Definition at line 2435 of file STM8L10x.h.

5.2.2.102 _COMP_COMP1_CH1

```
#define _COMP_COMP1_CH1 ((uint8_t) (0x01 << 0))
```

COMP Comparator 1 switch 1 enable [0] (in _COMP_CCS)

Definition at line 2448 of file STM8L10x.h.

5.2.2.103 _COMP_COMP1_CH2

```
#define _COMP_COMP1_CH2 ((uint8_t) (0x01 << 1))
```

COMP Comparator 1 switch 2 enable [0] (in _COMP_CCS)

Definition at line 2449 of file STM8L10x.h.

5.2.2.104 _COMP_COMP1_CH3

```
#define _COMP_COMP1_CH3 ((uint8_t) (0x01 << 2))
```

COMP Comparator 1 switch 3 enable [0] (in _COMP_CCS)

Definition at line 2450 of file STM8L10x.h.

5.2.2.105 _COMP_COMP1_CH4

```
#define _COMP_COMP1_CH4 ((uint8_t) (0x01 << 3))
```

COMP Comparator 1 switch 4 enable [0] (in _COMP_CCS)

Definition at line 2451 of file STM8L10x.h.

5.2.2.106 _COMP_COMP1_EN

```
#define _COMP_COMP1_EN ((uint8_t) (0x01 << 1))
```

COMP First comparator enable [0] (in _COMP_CR)

Definition at line 2429 of file STM8L10x.h.

5.2.2.107 _COMP_COMP1_OUT

```
#define _COMP_COMP1_OUT ((uint8_t) (0x01 << 0))
```

COMP First comparator output [0] (in _COMP_CSR)

Definition at line 2439 of file STM8L10x.h.

5.2.2.108 _COMP_COMP2_CH1

```
#define _COMP_COMP2_CH1 ((uint8_t) (0x01 << 4))
```

COMP Comparator 2 switch 1 enable [0] (in _COMP_CCS)

Definition at line 2452 of file STM8L10x.h.

5.2.2.109 _COMP_COMP2_CH2

```
#define _COMP_COMP2_CH2 ((uint8_t) (0x01 << 5))
```

COMP Comparator 2 switch 2 enable [0] (in _COMP_CCS)

Definition at line 2453 of file STM8L10x.h.

5.2.2.110 _COMP_COMP2_CH3

```
#define _COMP_COMP2_CH3 ((uint8_t) (0x01 << 6))
```

COMP Comparator 2 switch 3 enable [0] (in _COMP_CCS)

Definition at line 2454 of file STM8L10x.h.

5.2.2.111 _COMP_COMP2_CH4

```
#define _COMP_COMP2_CH4 ((uint8_t) (0x01 << 7))
```

COMP Comparator 2 switch 4 enable [0] (in _COMP_CCS)

Definition at line 2455 of file STM8L10x.h.

5.2.2.112 _COMP_COMP2_EN

```
#define _COMP_COMP2_EN ((uint8_t) (0x01 << 2))
```

COMP Second comparator enable [0] (in _COMP_CR)

Definition at line 2430 of file STM8L10x.h.

5.2.2.113 _COMP_COMP2_OUT

```
#define _COMP_COMP2_OUT ((uint8_t) (0x01 << 1))
```

COMP Second comparator output [0] (in _COMP_CSR)

Definition at line 2440 of file STM8L10x.h.

5.2.2.114 _COMP_COMPREF

```
#define _COMP_COMPREF ((uint8_t) (0x01 << 3))
```

COMP Comparator reference [0] (in _COMP_CR)

Definition at line 2431 of file STM8L10x.h.

5.2.2.115 _COMP_CR

```
#define _COMP_CR _SFR(uint8_t, COMP_AddressBase+0x00)
```

Comparator control register.

Definition at line 2418 of file STM8L10x.h.

5.2.2.116 _COMP_CR_RESET_VALUE

```
#define _COMP_CR_RESET_VALUE ((uint8_t) 0x00)
```

Comparator control register reset value.

Definition at line 2423 of file STM8L10x.h.

5.2.2.117 _COMP_CSR

```
#define _COMP_CSR _SFR(uint8_t, COMP_AddressBase+0x01)
```

Comparator control status register.

Definition at line 2419 of file STM8L10x.h.

5.2.2.118 _COMP_CSR_RESET_VALUE

```
#define _COMP_CSR_RESET_VALUE ((uint8_t) 0x00)
```

Comparator control status register reset value.

Definition at line 2424 of file STM8L10x.h.

5.2.2.119 _COMP_IC1_BK

```
#define _COMP_IC1_BK ((uint8_t) (0x01 << 7))
```

COMP Input capture 1 / break selection [0] (in _COMP_CR)

Definition at line 2436 of file STM8L10x.h.

5.2.2.120 _COMP_ITEN1

```
#define _COMP_ITEN1 ((uint8_t) (0x01 << 5))
```

COMP First comparator interrupt enable [0] (in _COMP_CSR)

Definition at line 2443 of file STM8L10x.h.

5.2.2.121 _COMP_ITEN2

```
#define _COMP_ITEN2 ((uint8_t) (0x01 << 7))
```

COMP Second comparator interrupt enable [0] (in _COMP_CSR)

Definition at line 2445 of file STM8L10x.h.

5.2.2.122 _COMP_POL

```
#define _COMP_POL ((uint8_t) (0x01 << 4))
```

COMP Comparator polarity [0] (in _COMP_CR)

Definition at line 2432 of file STM8L10x.h.

5.2.2.123 _EXTI

```
#define _EXTI _SFR(EXTI_t, EXTI_AddressBase)
```

External interrupt struct/bit access.

Definition at line 565 of file STM8L10x.h.

5.2.2.124 _EXTI_CONF

```
#define _EXTI_CONF _SFR(uint8_t, EXTI_AddressBase+0x05)
```

External interrupt port selector (EXTI_CONF)

Definition at line 571 of file STM8L10x.h.

5.2.2.125 _EXTI_CONF_RESET_VALUE

```
#define _EXTI_CONF_RESET_VALUE ((uint8_t) 0x00)
```

External interrupt port selector reset value.

Definition at line 579 of file STM8L10x.h.

5.2.2.126 _EXTI_CR1

```
#define _EXTI_CR1 _SFR(uint8_t, EXTI_AddressBase+0x00)
```

External interrupt control register 1 (EXTI_CR1)

Definition at line 566 of file STM8L10x.h.

5.2.2.127 _EXTI_CR1_RESET_VALUE

```
#define _EXTI_CR1_RESET_VALUE ((uint8_t) 0x00)
```

External interrupt control register 1 reset value.

Definition at line 574 of file STM8L10x.h.

5.2.2.128 _EXTI_CR2

```
#define _EXTI_CR2 _SFR(uint8_t, EXTI_AddressBase+0x01)
```

External interrupt control register 2 (EXTI_CR2)

Definition at line 567 of file STM8L10x.h.

5.2.2.129 _EXTI_CR2_RESET_VALUE

```
#define _EXTI_CR2_RESET_VALUE ((uint8_t) 0x00)
```

External interrupt control register 2 reset value.

Definition at line 575 of file STM8L10x.h.

5.2.2.130 _EXTI_CR3

```
#define _EXTI_CR3 _SFR(uint8_t, EXTI_AddressBase+0x02)
```

External interrupt control register 3 (EXTI_CR2)

Definition at line 568 of file STM8L10x.h.

5.2.2.131 _EXTI_CR3_RESET_VALUE

```
#define _EXTI_CR3_RESET_VALUE ((uint8_t) 0x00)
```

External interrupt control register 3 reset value.

Definition at line 576 of file STM8L10x.h.

5.2.2.132 _EXTI_P0F

```
#define _EXTI_P0F ((uint8_t) (0x01 << 0))
```

Portx bit 0 external interrupt flag (in _EXTI_SR1)

Definition at line 619 of file STM8L10x.h.

5.2.2.133 _EXTI_P0IS

```
#define _EXTI_P0IS ((uint8_t) (0x03 << 0))
```

External interrupt sensitivity for Portx bit 0 [1:0] (in _EXTI_CR1)

Definition at line 582 of file STM8L10x.h.

5.2.2.134 _EXTI_P0IS0

```
#define _EXTI_P0IS0 ((uint8_t) (0x01 << 0))
```

External interrupt sensitivity for Portx bit 0 [0] (in _EXTI_CR1)

Definition at line 583 of file STM8L10x.h.

5.2.2.135 _EXTI_P0IS1

```
#define _EXTI_P0IS1 ((uint8_t) (0x01 << 1))
```

External interrupt sensitivity for Portx bit 0 [1] (in _EXTI_CR1)

Definition at line 584 of file STM8L10x.h.

5.2.2.136 _EXTI_P1F

```
#define _EXTI_P1F ((uint8_t) (0x02 << 0))
```

Portx bit 1 external interrupt flag (in _EXTI_SR1)

Definition at line 620 of file STM8L10x.h.

5.2.2.137 _EXTI_P1IS

```
#define _EXTI_P1IS ((uint8_t) (0x03 << 0))
```

External interrupt sensitivity for Portx bit 1 [1:0] (in _EXTI_CR1)

Definition at line 585 of file STM8L10x.h.

5.2.2.138 _EXTI_P1IS0

```
#define _EXTI_P1IS0 ((uint8_t) (0x01 << 0))
```

External interrupt sensitivity for Portx bit 1 [0] (in _EXTI_CR1)

Definition at line 586 of file STM8L10x.h.

5.2.2.139 _EXTI_P1IS1

```
#define _EXTI_P1IS1 ((uint8_t) (0x01 << 1))
```

External interrupt sensitivity for Portx bit 1 [1] (in _EXTI_CR1)

Definition at line 587 of file STM8L10x.h.

5.2.2.140 _EXTI_P2F

```
#define _EXTI_P2F ((uint8_t) (0x04 << 0))
```

Portx bit 2 external interrupt flag (in _EXTI_SR1)

Definition at line 621 of file STM8L10x.h.

5.2.2.141 _EXTI_P2IS

```
#define _EXTI_P2IS ((uint8_t) (0x03 << 0))
```

External interrupt sensitivity for Portx bit 2 [1:0] (in _EXTI_CR1)

Definition at line 588 of file STM8L10x.h.

5.2.2.142 _EXTI_P2IS0

```
#define _EXTI_P2IS0 ((uint8_t) (0x01 << 0))
```

External interrupt sensitivity for Portx bit 2 [0] (in _EXTI_CR1)

Definition at line 589 of file STM8L10x.h.

5.2.2.143 _EXTI_P2IS1

```
#define _EXTI_P2IS1 ((uint8_t) (0x01 << 1))
```

External interrupt sensitivity for Portx bit 2 [1] (in _EXTI_CR1)

Definition at line 590 of file STM8L10x.h.

5.2.2.144 _EXTI_P3F

```
#define _EXTI_P3F ((uint8_t) (0x08 << 0))
```

Portx bit 3 external interrupt flag (in _EXTI_SR1)

Definition at line 622 of file STM8L10x.h.

5.2.2.145 _EXTI_P3IS

```
#define _EXTI_P3IS ((uint8_t) (0x03 << 0))
```

External interrupt sensitivity for Portx bit 3 [1:0] (in _EXTI_CR1)

Definition at line 591 of file STM8L10x.h.

5.2.2.146 _EXTI_P3IS0

```
#define _EXTI_P3IS0 ((uint8_t) (0x01 << 0))
```

External interrupt sensitivity for Portx bit 3 [0] (in _EXTI_CR1)

Definition at line 592 of file STM8L10x.h.

5.2.2.147 _EXTI_P3IS1

```
#define _EXTI_P3IS1 ((uint8_t) (0x01 << 1))
```

External interrupt sensitivity for Portx bit 3 [1] (in _EXTI_CR1)

Definition at line 593 of file STM8L10x.h.

5.2.2.148 _EXTI_P4F

```
#define _EXTI_P4F ((uint8_t) (0x10 << 0))
```

Portx bit 4 external interrupt flag (in _EXTI_SR1)

Definition at line 623 of file STM8L10x.h.

5.2.2.149 _EXTI_P4IS

```
#define _EXTI_P4IS ((uint8_t) (0x03 << 0))
```

External interrupt sensitivity for Portx bit 4 [1:0] (in _EXTI_CR2)

Definition at line 596 of file STM8L10x.h.

5.2.2.150 _EXTI_P4IS0

```
#define _EXTI_P4IS0 ((uint8_t) (0x01 << 0))
```

External interrupt sensitivity for Portx bit 4 [0] (in _EXTI_CR2)

Definition at line 597 of file STM8L10x.h.

5.2.2.151 _EXTI_P4IS1

```
#define _EXTI_P4IS1 ((uint8_t) (0x01 << 1))
```

External interrupt sensitivity for Portx bit 4 [1] (in _EXTI_CR2)

Definition at line 598 of file STM8L10x.h.

5.2.2.152 _EXTI_P5F

```
#define _EXTI_P5F ((uint8_t) (0x20 << 0))
```

Portx bit 5 external interrupt flag (in _EXTI_SR1)

Definition at line 624 of file STM8L10x.h.

5.2.2.153 _EXTI_P5IS

```
#define _EXTI_P5IS ((uint8_t) (0x03 << 0))
```

External interrupt sensitivity for Portx bit 5 [1:0] (in _EXTI_CR2)

Definition at line 599 of file STM8L10x.h.

5.2.2.154 _EXTI_P5IS0

```
#define _EXTI_P5IS0 ((uint8_t) (0x01 << 0))
```

External interrupt sensitivity for Portx bit 5 [0] (in _EXTI_CR2)

Definition at line 600 of file STM8L10x.h.

5.2.2.155 _EXTI_P5IS1

```
#define _EXTI_P5IS1 ((uint8_t) (0x01 << 1))
```

External interrupt sensitivity for Portx bit 5 [1] (in _EXTI_CR2)

Definition at line 601 of file STM8L10x.h.

5.2.2.156 _EXTI_P6F

```
#define _EXTI_P6F ((uint8_t) (0x40 << 0))
```

Portx bit 6 external interrupt flag (in _EXTI_SR1)

Definition at line 625 of file STM8L10x.h.

5.2.2.157 _EXTI_P6IS

```
#define _EXTI_P6IS ((uint8_t) (0x03 << 0))
```

External interrupt sensitivity for Portx bit 6 [1:0] (in _EXTI_CR2)

Definition at line 602 of file STM8L10x.h.

5.2.2.158 _EXTI_P6IS0

```
#define _EXTI_P6IS0 ((uint8_t) (0x01 << 0))
```

External interrupt sensitivity for Portx bit 6 [0] (in _EXTI_CR2)

Definition at line 603 of file STM8L10x.h.

5.2.2.159 _EXTI_P6IS1

```
#define _EXTI_P6IS1 ((uint8_t) (0x01 << 1))
```

External interrupt sensitivity for Portx bit 6 [1] (in _EXTI_CR2)

Definition at line 604 of file STM8L10x.h.

5.2.2.160 _EXTI_P7F

```
#define _EXTI_P7F ((uint8_t) (0x80 << 0))
```

Portx bit 7 external interrupt flag (in _EXTI_SR1)

Definition at line 626 of file STM8L10x.h.

5.2.2.161 _EXTI_P7IS

```
#define _EXTI_P7IS ((uint8_t) (0x03 << 0))
```

External interrupt sensitivity for Portx bit 7 [1:0] (in _EXTI_CR2)

Definition at line 605 of file STM8L10x.h.

5.2.2.162 _EXTI_P7IS0

```
#define _EXTI_P7IS0 ((uint8_t) (0x01 << 0))
```

External interrupt sensitivity for Portx bit 7 [0] (in _EXTI_CR2)

Definition at line 606 of file STM8L10x.h.

5.2.2.163 _EXTI_P7IS1

```
#define _EXTI_P7IS1 ((uint8_t) (0x01 << 1))
```

External interrupt sensitivity for Portx bit 7 [1] (in _EXTI_CR2)

Definition at line 607 of file STM8L10x.h.

5.2.2.164 _EXTI_PBF

```
#define _EXTI_PBF ((uint8_t) (0x01 << 0))
```

Port B external interrupt flag (in _EXTI_SR2)

Definition at line 629 of file STM8L10x.h.

5.2.2.165 _EXTI_PBHIS

```
#define _EXTI_PBHIS ((uint8_t) (0x02 << 0))
```

Port B, pins 4..7 external interrupt select (in _EXTI_CONF)

Definition at line 635 of file STM8L10x.h.

5.2.2.166 _EXTI_PBIS

```
#define _EXTI_PBIS ((uint8_t) (0x03 << 0))
```

Port B external interrupt sensitivity bits [1:0] (in _EXTI_CR3)

Definition at line 610 of file STM8L10x.h.

5.2.2.167 _EXTI_PBIS0

```
#define _EXTI_PBIS0 ((uint8_t) (0x01 << 0))
```

Port B external interrupt sensitivity bits [0] (in _EXTI_CR3)

Definition at line 611 of file STM8L10x.h.

5.2.2.168 _EXTI_PBIS1

```
#define _EXTI_PBIS1 ((uint8_t) (0x01 << 1))
```

Port B external interrupt sensitivity bits [1] (in _EXTI_CR3)

Definition at line 612 of file STM8L10x.h.

5.2.2.169 _EXTI_PBLIS

```
#define _EXTI_PBLIS ((uint8_t) (0x01 << 0))
```

Port B, pins 0..3 external interrupt select (in _EXTI_CONF)

Definition at line 634 of file STM8L10x.h.

5.2.2.170 _EXTI_PDF

```
#define _EXTI_PDF ((uint8_t) (0x02 << 0))
```

Port D external interrupt flag (in _EXTI_SR2)

Definition at line 630 of file STM8L10x.h.

5.2.2.171 _EXTI_PDHIS

```
#define _EXTI_PDHIS ((uint8_t) (0x02 << 0))
```

Port D, pins 4..7 external interrupt select (in _EXTI_CONF)

Definition at line 637 of file STM8L10x.h.

5.2.2.172 _EXTI_PDIS

```
#define _EXTI_PDIS ((uint8_t) (0x03 << 0))
```

Port D external interrupt sensitivity bits [1:0] (in _EXTI_CR3)

Definition at line 613 of file STM8L10x.h.

5.2.2.173 _EXTI_PDIS0

```
#define _EXTI_PDIS0 ((uint8_t) (0x01 << 0))
```

Port D external interrupt sensitivity bits [0] (in _EXTI_CR3)

Definition at line 614 of file STM8L10x.h.

5.2.2.174 _EXTI_PDIS1

```
#define _EXTI_PDIS1 ((uint8_t) (0x01 << 1))
```

Port D external interrupt sensitivity bits [1] (in _EXTI_CR3)

Definition at line 615 of file STM8L10x.h.

5.2.2.175 _EXTI_PDLIS

```
#define _EXTI_PDLIS ((uint8_t) (0x01 << 0))
```

Port D, pins 0..3 external interrupt select (in _EXTI_CONF)

Definition at line 636 of file STM8L10x.h.

5.2.2.176 _EXTI_SR1

```
#define _EXTI_SR1 _SFR(uint8_t, EXTI_AddressBase+0x03)
```

External interrupt status register 1 (EXTI_SR1)

Definition at line 569 of file STM8L10x.h.

5.2.2.177 _EXTI_SR1_RESET_VALUE

```
#define _EXTI_SR1_RESET_VALUE ((uint8_t) 0x00)
```

External interrupt status register 1 reset value.

Definition at line 577 of file STM8L10x.h.

5.2.2.178 _EXTI_SR2

```
#define _EXTI_SR2 _SFR(uint8_t, EXTI_AddressBase+0x04)
```

External interrupt status register 2 (EXTI_SR2)

Definition at line 570 of file STM8L10x.h.

5.2.2.179 _EXTI_SR2_RESET_VALUE

```
#define _EXTI_SR2_RESET_VALUE ((uint8_t) 0x00)
```

External interrupt status register 2 reset value.

Definition at line 578 of file STM8L10x.h.

5.2.2.180 _FLASH

```
#define _FLASH _SFR(FLASH_t, FLASH_AddressBase)
```

Flash struct/bit access.

Definition at line 460 of file STM8L10x.h.

5.2.2.181 _FLASH_CR1

```
#define _FLASH_CR1 _SFR(uint8_t, FLASH_AddressBase+0x00)
```

Flash control register 1 (FLASH_CR1)

Definition at line 461 of file STM8L10x.h.

5.2.2.182 _FLASH_CR1_RESET_VALUE

```
#define _FLASH_CR1_RESET_VALUE ((uint8_t) 0x00)
```

Flash control register 1 reset value.

Definition at line 468 of file STM8L10x.h.

5.2.2.183 _FLASH_CR2

```
#define _FLASH_CR2 _SFR(uint8_t, FLASH_AddressBase+0x01)
```

Flash control register 2 (FLASH_CR2)

Definition at line 462 of file STM8L10x.h.

5.2.2.184 _FLASH_CR2_RESET_VALUE

```
#define _FLASH_CR2_RESET_VALUE ((uint8_t) 0x00)
```

Flash control register 2 reset value.

Definition at line 469 of file STM8L10x.h.

5.2.2.185 _FLASH_DUKR

```
#define _FLASH_DUKR _SFR(uint8_t, FLASH_AddressBase+0x03)
```

Data EEPROM unprotection key register (FLASH_DUKR)

Definition at line 464 of file STM8L10x.h.

5.2.2.186 _FLASH_DUKR_RESET_VALUE

```
#define _FLASH_DUKR_RESET_VALUE ((uint8_t) 0x00)
```

Data EEPROM unprotection key reset value.

Definition at line 471 of file STM8L10x.h.

5.2.2.187 _FLASH_DUL

```
#define _FLASH_DUL ((uint8_t) (0x01 << 3))
```

Data EEPROM area unlocked flag [0] (in _FLASH_IAPSR)

Definition at line 491 of file STM8L10x.h.

5.2.2.188 _FLASH_EOP

```
#define _FLASH_EOP ((uint8_t) (0x01 << 2))
```

End of programming (write or erase operation) flag [0] (in _FLASH_IAPSR)

Definition at line 490 of file STM8L10x.h.

5.2.2.189 _FLASH_ERASE

```
#define _FLASH_ERASE ((uint8_t) (0x01 << 5))
```

Block erasing [0] (in _FLASH_CR2 and _FLASH_NCR2)

Definition at line 483 of file STM8L10x.h.

5.2.2.190 _FLASH_FIX

```
#define _FLASH_FIX ((uint8_t) (0x01 << 0))
```

Fixed Byte programming time [0] (in _FLASH_CR1)

Definition at line 475 of file STM8L10x.h.

5.2.2.191 _FLASH_FPRG

```
#define _FLASH_FPRG ((uint8_t) (0x01 << 4))
```

Fast block programming [0] (in _FLASH_CR2 and _FLASH_NCR2)

Definition at line 482 of file STM8L10x.h.

5.2.2.192 _FLASH_IAPSR

```
#define _FLASH_IAPSR _SFR(uint8_t, FLASH_AddressBase+0x04)
```

Flash status register (FLASH_IAPSR)

Definition at line 465 of file STM8L10x.h.

5.2.2.193 _FLASH_IAPSR_RESET_VALUE

```
#define _FLASH_IAPSR_RESET_VALUE ((uint8_t) 0x40)
```

Flash status register reset value.

Definition at line 472 of file STM8L10x.h.

5.2.2.194 _FLASH_IE

```
#define _FLASH_IE ((uint8_t) (0x01 << 1))
```

Flash Interrupt enable [0] (in _FLASH_CR1)

Definition at line 476 of file STM8L10x.h.

5.2.2.195 _FLASH_OPT

```
#define _FLASH_OPT ((uint8_t) (0x01 << 7))
```

Write option bytes [0] (in _FLASH_CR2 and _FLASH_NCR2)

Definition at line 485 of file STM8L10x.h.

5.2.2.196 _FLASH_PRG

```
#define _FLASH_PRG ((uint8_t) (0x01 << 0))
```

Standard block programming [0] (in _FLASH_CR2 and _FLASH_NCR2)

Definition at line 480 of file STM8L10x.h.

5.2.2.197 _FLASH_PUKR

```
#define _FLASH_PUKR _SFR(uint8_t, FLASH_AddressBase+0x02)
```

Flash program memory unprotecting key register (FLASH_PUKR)

Definition at line 463 of file STM8L10x.h.

5.2.2.198 _FLASH_PUKR_RESET_VALUE

```
#define _FLASH_PUKR_RESET_VALUE ((uint8_t) 0x00)
```

Flash program memory unprotecting key reset value.

Definition at line 470 of file STM8L10x.h.

5.2.2.199 _FLASH_PUL

```
#define _FLASH_PUL ((uint8_t) (0x01 << 1))
```

Flash Program memory unlocked flag [0] (in _FLASH_IAPSR)

Definition at line 489 of file STM8L10x.h.

5.2.2.200 _FLASH_WPRG

```
#define _FLASH_WPRG ((uint8_t) (0x01 << 6))
```

Word programming [0] (in _FLASH_CR2 and _FLASH_NCR2)

Definition at line 484 of file STM8L10x.h.

5.2.2.201 _FLASH_WR_PG_DIS

```
#define _FLASH_WR_PG_DIS ((uint8_t) (0x01 << 0))
```

Write attempted to protected page flag [0] (in _FLASH_IAPSR)

Definition at line 488 of file STM8L10x.h.

5.2.2.202 _GPIO_CR1_RESET_VALUE

```
#define _GPIO_CR1_RESET_VALUE ((uint8_t) 0x00)
```

port control register 1 reset value

Definition at line 392 of file STM8L10x.h.

5.2.2.203 _GPIO_CR2_RESET_VALUE

```
#define _GPIO_CR2_RESET_VALUE ((uint8_t) 0x00)
```

port control register 2 reset value

Definition at line 393 of file STM8L10x.h.

5.2.2.204 _GPIO_DDR_RESET_VALUE

```
#define _GPIO_DDR_RESET_VALUE ((uint8_t) 0x00)
```

port direction register reset value

Definition at line 391 of file STM8L10x.h.

5.2.2.205 _GPIO_ODR_RESET_VALUE

```
#define _GPIO_ODR_RESET_VALUE ((uint8_t) 0x00)
```

port output register reset value

Definition at line 390 of file STM8L10x.h.

5.2.2.206 _GPIO_PIN0

```
#define _GPIO_PIN0 ((uint8_t) (0x01 << 0))
```

port bit mask for pin 0 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 396 of file STM8L10x.h.

5.2.2.207 _GPIO_PIN1

```
#define _GPIO_PIN1 ((uint8_t) (0x01 << 1))
```

port bit mask for pin 1 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 397 of file STM8L10x.h.

5.2.2.208 _GPIO_PIN2

```
#define _GPIO_PIN2 ((uint8_t) (0x01 << 2))
```

port bit mask for pin 2 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 398 of file STM8L10x.h.

5.2.2.209 _GPIO_PIN3

```
#define _GPIO_PIN3 ((uint8_t) (0x01 << 3))
```

port bit mask for pin 3 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 399 of file STM8L10x.h.

5.2.2.210 _GPIO_PIN4

```
#define _GPIO_PIN4 ((uint8_t) (0x01 << 4))
```

port bit mask for pin 4 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 400 of file STM8L10x.h.

5.2.2.211 _GPIO_PIN5

```
#define _GPIO_PIN5 ((uint8_t) (0x01 << 5))
```

port bit mask for pin 5 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 401 of file STM8L10x.h.

5.2.2.212 _GPIO_PIN6

```
#define _GPIO_PIN6 ((uint8_t) (0x01 << 6))
```

port bit mask for pin 6 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 402 of file STM8L10x.h.

5.2.2.213 _GPIO_PIN7

```
#define _GPIO_PIN7 ((uint8_t) (0x01 << 7))
```

port bit mask for pin 7 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)

Definition at line 403 of file STM8L10x.h.

5.2.2.214 _GPIOA

```
#define _GPIOA _SFR(PORT_t, PORTA_AddressBase)
```

port A struct/bit access

Definition at line 351 of file STM8L10x.h.

5.2.2.215 _GPIOA_CR1

```
#define _GPIOA_CR1 _SFR(uint8_t, PORTA_AddressBase+0x03)
```

port A control register 1

Definition at line 355 of file STM8L10x.h.

5.2.2.216 _GPIOA_CR2

```
#define _GPIOA_CR2 _SFR(uint8_t, PORTA_AddressBase+0x04)
```

port A control register 2

Definition at line 356 of file STM8L10x.h.

5.2.2.217 _GPIOA_DDR

```
#define _GPIOA_DDR _SFR(uint8_t, PORTA_AddressBase+0x02)
```

port A direction register

Definition at line 354 of file STM8L10x.h.

5.2.2.218 _GPIOA_IDR

```
#define _GPIOA_IDR _SFR(uint8_t, PORTA_AddressBase+0x01)
```

port A input register

Definition at line 353 of file STM8L10x.h.

5.2.2.219 _GPIOA_ODR

```
#define _GPIOA_ODR _SFR(uint8_t, PORTA_AddressBase+0x00)
```

port A output register

Definition at line 352 of file STM8L10x.h.

5.2.2.220 _GPIOB

```
#define _GPIOB _SFR(PORT_t, PORTB_AddressBase)
```

port B struct/bit access

Definition at line 361 of file STM8L10x.h.

5.2.2.221 _GPIOB_CR1

```
#define _GPIOB_CR1 _SFR(uint8_t, PORTB_AddressBase+0x03)
```

port B control register 1

Definition at line 365 of file STM8L10x.h.

5.2.2.222 _GPIOB_CR2

```
#define _GPIOB_CR2 _SFR(uint8_t, PORTB_AddressBase+0x04)
```

port B control register 2

Definition at line 366 of file STM8L10x.h.

5.2.2.223 _GPIOB_DDR

```
#define _GPIOB_DDR _SFR(uint8_t, PORTB_AddressBase+0x02)
```

port B direction register

Definition at line 364 of file STM8L10x.h.

5.2.2.224 _GPIOB_IDR

```
#define _GPIOB_IDR _SFR(uint8_t, PORTB_AddressBase+0x01)
```

port B input register

Definition at line 363 of file STM8L10x.h.

5.2.2.225 _GPIOB_ODR

```
#define _GPIOB_ODR _SFR(uint8_t, PORTB_AddressBase+0x00)
```

port B output register

Definition at line 362 of file STM8L10x.h.

5.2.2.226 _GPIOC

```
#define _GPIOC _SFR(PORT_t, PORTC_AddressBase)
```

port C struct/bit access

Definition at line 371 of file STM8L10x.h.

5.2.2.227 _GPIOC_CR1

```
#define _GPIOC_CR1 _SFR(uint8_t, PORTC_AddressBase+0x03)
```

port C control register 1

Definition at line 375 of file STM8L10x.h.

5.2.2.228 _GPIOC_CR2

```
#define _GPIOC_CR2 _SFR(uint8_t, PORTC_AddressBase+0x04)
```

port C control register 2

Definition at line 376 of file STM8L10x.h.

5.2.2.229 _GPIOC_DDR

```
#define _GPIOC_DDR _SFR(uint8_t, PORTC_AddressBase+0x02)
```

port C direction register

Definition at line 374 of file STM8L10x.h.

5.2.2.230 _GPIOC_IDR

```
#define _GPIOC_IDR _SFR(uint8_t, PORTC_AddressBase+0x01)
```

port C input register

Definition at line 373 of file STM8L10x.h.

5.2.2.231 _GPIOC_ODR

```
#define _GPIOC_ODR _SFR(uint8_t, PORTC_AddressBase+0x00)
```

port C output register

Definition at line 372 of file STM8L10x.h.

5.2.2.232 _GPIOD

```
#define _GPIOD _SFR(PORT_t, PORTD_AddressBase)
```

port D struct/bit access

Definition at line 381 of file STM8L10x.h.

5.2.2.233 _GPIOD_CR1

```
#define _GPIOD_CR1 _SFR(uint8_t, PORTD_AddressBase+0x03)
```

port D control register 1

Definition at line 385 of file STM8L10x.h.

5.2.2.234 _GPIOD_CR2

```
#define _GPIOD_CR2 _SFR(uint8_t, PORTD_AddressBase+0x04)
```

port D control register 2

Definition at line 386 of file STM8L10x.h.

5.2.2.235 _GPIOD_DDR

```
#define _GPIOD_DDR _SFR(uint8_t, PORTD_AddressBase+0x02)
```

port D direction register

Definition at line 384 of file STM8L10x.h.

5.2.2.236 _GPIOD_IDR

```
#define _GPIOD_IDR _SFR(uint8_t, PORTD_AddressBase+0x01)
```

port D input register

Definition at line 383 of file STM8L10x.h.

5.2.2.237 _GPIOD_ODR

```
#define _GPIOD_ODR _SFR(uint8_t, PORTD_AddressBase+0x00)
```

port D output register

Definition at line 382 of file STM8L10x.h.

5.2.2.238 _I2C

```
#define _I2C _SFR(I2C_t, I2C_AddressBase)
```

register for SPI control

I2C struct/bit access

Definition at line 1193 of file STM8L10x.h.

5.2.2.239 _I2C_ACK

```
#define _I2C_ACK ((uint8_t) (0x01 << 2))
```

I2C Acknowledge enable [0] (in _I2C_CR2)

Definition at line 1234 of file STM8L10x.h.

5.2.2.240 _I2C_ADD0

```
#define _I2C_ADD0 ((uint8_t) (0x01 << 0))
```

I2C Interface address [0] (in 10-bit address mode) (in _I2C_OARL)

Definition at line 1250 of file STM8L10x.h.

5.2.2.241 _I2C_ADD1

```
#define _I2C_ADD1 ((uint8_t) (0x01 << 1))
```

I2C Interface address [1] (in _I2C_OARL)

Definition at line 1251 of file STM8L10x.h.

5.2.2.242 _I2C_ADD10

```
#define _I2C_ADD10 ((uint8_t) (0x01 << 3))
```

I2C 10-bit header sent (Master mode) [0] (in _I2C_SR1)

Definition at line 1272 of file STM8L10x.h.

5.2.2.243 _I2C_ADD2

```
#define _I2C_ADD2 ((uint8_t) (0x01 << 2))
```

I2C Interface address [2] (in _I2C_OARL)

Definition at line 1252 of file STM8L10x.h.

5.2.2.244 _I2C_ADD3

```
#define _I2C_ADD3 ((uint8_t) (0x01 << 3))
```

I2C Interface address [3] (in _I2C_OARL)

Definition at line 1253 of file STM8L10x.h.

5.2.2.245 _I2C_ADD4

```
#define _I2C_ADD4 ((uint8_t) (0x01 << 4))
```

I2C Interface address [4] (in _I2C_OARL)

Definition at line 1254 of file STM8L10x.h.

5.2.2.246 _I2C_ADD5

```
#define _I2C_ADD5 ((uint8_t) (0x01 << 5))
```

I2C Interface address [5] (in _I2C_OARL)

Definition at line 1255 of file STM8L10x.h.

5.2.2.247 _I2C_ADD6

```
#define _I2C_ADD6 ((uint8_t) (0x01 << 6))
```

I2C Interface address [6] (in _I2C_OARL)

Definition at line 1256 of file STM8L10x.h.

5.2.2.248 _I2C_ADD7

```
#define _I2C_ADD7 ((uint8_t) (0x01 << 7))
```

I2C Interface address [7] (in _I2C_OARL)

Definition at line 1257 of file STM8L10x.h.

5.2.2.249 _I2C_ADD8

```
#define _I2C_ADD8 ((uint8_t) (0x01 << 1))
```

I2C Interface address [8] (in _I2C_OARH)

Definition at line 1262 of file STM8L10x.h.

5.2.2.250 _I2C_ADD9

```
#define _I2C_ADD9 ((uint8_t) (0x01 << 2))
```

I2C Interface address [9] (in _I2C_OARH)

Definition at line 1263 of file STM8L10x.h.

5.2.2.251 _I2C_ADD_8_9

```
#define _I2C_ADD_8_9 ((uint8_t) (0x03 << 1))
```

I2C Interface address [9:8] (in 10-bit address mode) (in _I2C_OARH)

Definition at line 1261 of file STM8L10x.h.

5.2.2.252 _I2C_ADDCONF

```
#define _I2C_ADDCONF ((uint8_t) (0x01 << 6))
```

I2C Address mode configuration [0] (in _I2C_OARH)

Definition at line 1265 of file STM8L10x.h.

5.2.2.253 _I2C_ADDMODE

```
#define _I2C_ADDMODE ((uint8_t) (0x01 << 7))
```

I2C 7-/10-bit addressing mode (Slave mode) [0] (in _I2C_OARH)

Definition at line 1266 of file STM8L10x.h.

5.2.2.254 _I2C_ADDR

```
#define _I2C_ADDR ((uint8_t) (0x01 << 1))
```

I2C Address sent (Master mode) / matched (Slave mode) [0] (in _I2C_SR1)

Definition at line 1270 of file STM8L10x.h.

5.2.2.255 _I2C_AF

```
#define _I2C_AF ((uint8_t) (0x01 << 2))
```

I2C Acknowledge failure [0] (in _I2C_SR2)

Definition at line 1281 of file STM8L10x.h.

5.2.2.256 _I2C_ARLO

```
#define _I2C_ARLO ((uint8_t) (0x01 << 1))
```

I2C Arbitration lost (Master mode) [0] (in _I2C_SR2)

Definition at line 1280 of file STM8L10x.h.

5.2.2.257 _I2C_BERR

```
#define _I2C_BERR ((uint8_t) (0x01 << 0))
```

I2C Bus error [0] (in _I2C_SR2)

Definition at line 1279 of file STM8L10x.h.

5.2.2.258 _I2C_BTF

```
#define _I2C_BTF ((uint8_t) (0x01 << 2))
```

I2C Byte transfer finished [0] (in _I2C_SR1)

Definition at line 1271 of file STM8L10x.h.

5.2.2.259 _I2C_BUSY

```
#define _I2C_BUSY ((uint8_t) (0x01 << 1))
```

I2C Bus busy [0] (in _I2C_SR3)

Definition at line 1289 of file STM8L10x.h.

5.2.2.260 _I2C_CCR

```
#define _I2C_CCR ((uint8_t) (0x0F << 0))
```

I2C Clock control register (Master mode) [3:0] (in _I2C_CCRH)

Definition at line 1303 of file STM8L10x.h.

5.2.2.261 _I2C_CCR0

```
#define _I2C_CCR0 ((uint8_t) (0x01 << 0))
```

I2C Clock control register (Master mode) [0] (in _I2C_CCRH)

Definition at line 1304 of file STM8L10x.h.

5.2.2.262 _I2C_CCR1

```
#define _I2C_CCR1 ((uint8_t) (0x01 << 1))
```

I2C Clock control register (Master mode) [1] (in _I2C_CCRH)

Definition at line 1305 of file STM8L10x.h.

5.2.2.263 _I2C_CCR2

```
#define _I2C_CCR2 ((uint8_t) (0x01 << 2))
```

I2C Clock control register (Master mode) [2] (in _I2C_CCRH)

Definition at line 1306 of file STM8L10x.h.

5.2.2.264 _I2C_CCR3

```
#define _I2C_CCR3 ((uint8_t) (0x01 << 3))
```

I2C Clock control register (Master mode) [3] (in _I2C_CCRH)

Definition at line 1307 of file STM8L10x.h.

5.2.2.265 _I2C_CCRH

```
#define _I2C_CCRH _SFR(uint8_t, I2C_AddressBase+0x0C)
```

I2C Clock control register high byte.

Definition at line 1206 of file STM8L10x.h.

5.2.2.266 _I2C_CCRH_RESET_VALUE

```
#define _I2C_CCRH_RESET_VALUE ((uint8_t) 0x00)
```

I2C Clock control register high byte reset value.

Definition at line 1222 of file STM8L10x.h.

5.2.2.267 _I2C_CCRL

```
#define _I2C_CCRL _SFR(uint8_t, I2C_AddressBase+0x0B)
```

I2C Clock control register low byte.

Definition at line 1205 of file STM8L10x.h.

5.2.2.268 _I2C_CCRL_RESET_VALUE

```
#define _I2C_CCRL_RESET_VALUE ((uint8_t) 0x00)
```

I2C Clock control register low byte reset value.

Definition at line 1221 of file STM8L10x.h.

5.2.2.269 _I2C_CR1

```
#define _I2C_CR1 _SFR(uint8_t, I2C_AddressBase+0x00)
```

I2C Control register 1.

Definition at line 1194 of file STM8L10x.h.

5.2.2.270 _I2C_CR1_RESET_VALUE

```
#define _I2C_CR1_RESET_VALUE ((uint8_t) 0x00)
```

I2C Control register 1 reset value.

Definition at line 1211 of file STM8L10x.h.

5.2.2.271 _I2C_CR2

```
#define _I2C_CR2 _SFR(uint8_t, I2C_AddressBase+0x01)
```

I2C Control register 2.

Definition at line 1195 of file STM8L10x.h.

5.2.2.272 _I2C_CR2_RESET_VALUE

```
#define _I2C_CR2_RESET_VALUE ((uint8_t) 0x00)
```

I2C Control register 2 reset value.

Definition at line 1212 of file STM8L10x.h.

5.2.2.273 _I2C_DR

```
#define _I2C_DR _SFR(uint8_t, I2C_AddressBase+0x06)
```

I2C data register.

Definition at line 1200 of file STM8L10x.h.

5.2.2.274 _I2C_DR_RESET_VALUE

```
#define _I2C_DR_RESET_VALUE ((uint8_t) 0x00)
```

I2C data register reset value.

Definition at line 1216 of file STM8L10x.h.

5.2.2.275 _I2C_DUALF

```
#define _I2C_DUALF ((uint8_t) (0x01 << 7))
```

Dual flag (Slave mode) [0] (in _I2C_SR3)

Definition at line 1294 of file STM8L10x.h.

5.2.2.276 _I2C_DUTY

```
#define _I2C_DUTY ((uint8_t) (0x01 << 6))
```

I2C Fast mode duty cycle [0] (in _I2C_CCRH)

Definition at line 1309 of file STM8L10x.h.

5.2.2.277 _I2C_ENGC

```
#define _I2C_ENGC ((uint8_t) (0x01 << 6))
```

I2C General call enable [0] (in _I2C_CR1)

Definition at line 1228 of file STM8L10x.h.

5.2.2.278 _I2C_FREQ

```
#define _I2C_FREQ ((uint8_t) (0x3F << 0))
```

I2C Peripheral clock frequency [5:0] (in _I2C_FREQR)

Definition at line 1240 of file STM8L10x.h.

5.2.2.279 _I2C_FREQ0

```
#define _I2C_FREQ0 ((uint8_t) (0x01 << 0))
```

I2C Peripheral clock frequency [0] (in _I2C_FREQR)

Definition at line 1241 of file STM8L10x.h.

5.2.2.280 _I2C_FREQ1

```
#define _I2C_FREQ1 ((uint8_t) (0x01 << 1))
```

I2C Peripheral clock frequency [1] (in _I2C_FREQR)

Definition at line 1242 of file STM8L10x.h.

5.2.2.281 _I2C_FREQ2

```
#define _I2C_FREQ2 ((uint8_t) (0x01 << 2))
```

I2C Peripheral clock frequency [2] (in _I2C_FREQR)

Definition at line 1243 of file STM8L10x.h.

5.2.2.282 _I2C_FREQ3

```
#define _I2C_FREQ3 ((uint8_t) (0x01 << 3))
```

I2C Peripheral clock frequency [3] (in _I2C_FREQR)

Definition at line 1244 of file STM8L10x.h.

5.2.2.283 _I2C_FREQ4

```
#define _I2C_FREQ4 ((uint8_t) (0x01 << 4))
```

I2C Peripheral clock frequency [4] (in _I2C_FREQR)

Definition at line 1245 of file STM8L10x.h.

5.2.2.284 _I2C_FREQ5

```
#define _I2C_FREQ5 ((uint8_t) (0x01 << 5))
```

I2C Peripheral clock frequency [5] (in _I2C_FREQR)

Definition at line 1246 of file STM8L10x.h.

5.2.2.285 _I2C_FREQR

```
#define _I2C_FREQR _SFR(uint8_t, I2C_AddressBase+0x02)
```

I2C Frequency register.

Definition at line 1196 of file STM8L10x.h.

5.2.2.286 _I2C_FREQR_RESET_VALUE

```
#define _I2C_FREQR_RESET_VALUE ((uint8_t) 0x00)
```

I2C Frequency register reset value.

Definition at line 1213 of file STM8L10x.h.

5.2.2.287 _I2C_FS

```
#define _I2C_FS ((uint8_t) (0x01 << 7))
```

I2C Master mode selection [0] (in _I2C_CCRH)

Definition at line 1310 of file STM8L10x.h.

5.2.2.288 _I2C_GENCALL

```
#define _I2C_GENCALL ((uint8_t) (0x01 << 4))
```

I2C General call header (Slave mode) [0] (in _I2C_SR3)

Definition at line 1292 of file STM8L10x.h.

5.2.2.289 _I2C_ITBUFEN

```
#define _I2C_ITBUFEN ((uint8_t) (0x01 << 2))
```

I2C Buffer interrupt enable [0] (in _I2C_ITR)

Definition at line 1299 of file STM8L10x.h.

5.2.2.290 _I2C_ITERREN

```
#define _I2C_ITERREN ((uint8_t) (0x01 << 0))
```

I2C Error interrupt enable [0] (in _I2C_ITR)

Definition at line 1297 of file STM8L10x.h.

5.2.2.291 _I2C_ITEVTEN

```
#define _I2C_ITEVTEN ((uint8_t) (0x01 << 1))
```

I2C Event interrupt enable [0] (in _I2C_ITR)

Definition at line 1298 of file STM8L10x.h.

5.2.2.292 _I2C_ITR

```
#define _I2C_ITR _SFR(uint8_t, I2C_AddressBase+0x0A)
```

I2C Interrupt register.

Definition at line 1204 of file STM8L10x.h.

5.2.2.293 _I2C_ITR_RESET_VALUE

```
#define _I2C_ITR_RESET_VALUE ((uint8_t) 0x00)
```

I2C Interrupt register reset value.

Definition at line 1220 of file STM8L10x.h.

5.2.2.294 _I2C_MSL

```
#define _I2C_MSL ((uint8_t) (0x01 << 0))
```

I2C Master/Slave [0] (in _I2C_SR3)

Definition at line 1288 of file STM8L10x.h.

5.2.2.295 _I2C_NOSTRETCH

```
#define _I2C_NOSTRETCH ((uint8_t) (0x01 << 7))
```

I2C Clock stretching disable (Slave mode) [0] (in _I2C_CR1)

Definition at line 1229 of file STM8L10x.h.

5.2.2.296 _I2C_OARH

```
#define _I2C_OARH _SFR(uint8_t, I2C_AddressBase+0x04)
```

I2C own address register high byte.

Definition at line 1198 of file STM8L10x.h.

5.2.2.297 _I2C_OARH_RESET_VALUE

```
#define _I2C_OARH_RESET_VALUE ((uint8_t) 0x00)
```

I2C own address register high byte reset value.

Definition at line 1215 of file STM8L10x.h.

5.2.2.298 _I2C_OARL

```
#define _I2C_OARL _SFR(uint8_t, I2C_AddressBase+0x03)
```

I2C own address register low byte.

Definition at line 1197 of file STM8L10x.h.

5.2.2.299 _I2C_OARL_RESET_VALUE

```
#define _I2C_OARL_RESET_VALUE ((uint8_t) 0x00)
```

I2C own address register low byte reset value.

Definition at line 1214 of file STM8L10x.h.

5.2.2.300 _I2C_OVR

```
#define _I2C_OVR ((uint8_t) (0x01 << 3))
```

I2C Overrun/underrun [0] (in _I2C_SR2)

Definition at line 1282 of file STM8L10x.h.

5.2.2.301 _I2C_PE

```
#define _I2C_PE ((uint8_t) (0x01 << 0))
```

I2C Peripheral enable [0] (in _I2C_CR1)

Definition at line 1226 of file STM8L10x.h.

5.2.2.302 _I2C_POS

```
#define _I2C_POS ((uint8_t) (0x01 << 3))
```

I2C Acknowledge position (for data reception) [0] (in _I2C_CR2)

Definition at line 1235 of file STM8L10x.h.

5.2.2.303 _I2C_RXNE

```
#define _I2C_RXNE ((uint8_t) (0x01 << 6))
```

I2C Data register not empty (receivers) [0] (in _I2C_SR1)

Definition at line 1275 of file STM8L10x.h.

5.2.2.304 _I2C_SB

```
#define _I2C_SB ((uint8_t) (0x01 << 0))
```

I2C Start bit (Master mode) [0] (in _I2C_SR1)

Definition at line 1269 of file STM8L10x.h.

5.2.2.305 _I2C_SR1

```
#define _I2C_SR1 _SFR(uint8_t, I2C_AddressBase+0x07)
```

I2C Status register 1.

Definition at line 1201 of file STM8L10x.h.

5.2.2.306 _I2C_SR1_RESET_VALUE

```
#define _I2C_SR1_RESET_VALUE ((uint8_t) 0x00)
```

I2C Status register 1 reset value.

Definition at line 1217 of file STM8L10x.h.

5.2.2.307 _I2C_SR2

```
#define _I2C_SR2 _SFR(uint8_t, I2C_AddressBase+0x08)
```

I2C Status register 2.

Definition at line 1202 of file STM8L10x.h.

5.2.2.308 _I2C_SR2_RESET_VALUE

```
#define _I2C_SR2_RESET_VALUE ((uint8_t) 0x00)
```

I2C Status register 2 reset value.

Definition at line 1218 of file STM8L10x.h.

5.2.2.309 _I2C_SR3

```
#define _I2C_SR3 _SFR(uint8_t, I2C_AddressBase+0x09)
```

I2C Status register 3.

Definition at line 1203 of file STM8L10x.h.

5.2.2.310 _I2C_SR3_RESET_VALUE

```
#define _I2C_SR3_RESET_VALUE ((uint8_t) 0x00)
```

I2C Status register 3 reset value.

Definition at line 1219 of file STM8L10x.h.

5.2.2.311 _I2C_START

```
#define _I2C_START ((uint8_t) (0x01 << 0))
```

I2C Start generation [0] (in _I2C_CR2)

Definition at line 1232 of file STM8L10x.h.

5.2.2.312 _I2C_STOP

```
#define _I2C_STOP ((uint8_t) (0x01 << 1))
```

I2C Stop generation [0] (in _I2C_CR2)

Definition at line 1233 of file STM8L10x.h.

5.2.2.313 _I2C_STOPF

```
#define _I2C_STOPF ((uint8_t) (0x01 << 4))
```

I2C Stop detection (Slave mode) [0] (in _I2C_SR1)

Definition at line 1273 of file STM8L10x.h.

5.2.2.314 _I2C_SWRST

```
#define _I2C_SWRST ((uint8_t) (0x01 << 7))
```

I2C Software reset [0] (in _I2C_CR2)

Definition at line 1237 of file STM8L10x.h.

5.2.2.315 _I2C_TRA

```
#define _I2C_TRA ((uint8_t) (0x01 << 2))
```

I2C Transmitter/Receiver [0] (in _I2C_SR3)

Definition at line 1290 of file STM8L10x.h.

5.2.2.316 _I2C_TRISE

```
#define _I2C_TRISE ((uint8_t) (0x3F << 0))
```

I2C Maximum rise time (Master mode) [5:0] (in _I2C_TRISER)

Definition at line 1313 of file STM8L10x.h.

5.2.2.317 _I2C_TRISE0

```
#define _I2C_TRISE0 ((uint8_t) (0x01 << 0))
```

I2C Maximum rise time (Master mode) [0] (in _I2C_TRISER)

Definition at line 1314 of file STM8L10x.h.

5.2.2.318 _I2C_TRISE1

```
#define _I2C_TRISE1 ((uint8_t) (0x01 << 1))
```

I2C Maximum rise time (Master mode) [1] (in _I2C_TRISER)

Definition at line 1315 of file STM8L10x.h.

5.2.2.319 _I2C_TRISE2

```
#define _I2C_TRISE2 ((uint8_t) (0x01 << 2))
```

I2C Maximum rise time (Master mode) [2] (in _I2C_TRISER)

Definition at line 1316 of file STM8L10x.h.

5.2.2.320 _I2C_TRISE3

```
#define _I2C_TRISE3 ((uint8_t) (0x01 << 3))
```

I2C Maximum rise time (Master mode) [3] (in _I2C_TRISER)

Definition at line 1317 of file STM8L10x.h.

5.2.2.321 _I2C_TRISE4

```
#define _I2C_TRISE4 ((uint8_t) (0x01 << 4))
```

I2C Maximum rise time (Master mode) [4] (in _I2C_TRISER)

Definition at line 1318 of file STM8L10x.h.

5.2.2.322 _I2C_TRISE5

```
#define _I2C_TRISE5 ((uint8_t) (0x01 << 5))
```

I2C Maximum rise time (Master mode) [5] (in _I2C_TRISER)

Definition at line 1319 of file STM8L10x.h.

5.2.2.323 _I2C_TRISER

```
#define _I2C_TRISER _SFR(uint8_t, I2C_AddressBase+0x0D)
```

I2C rise time register.

Definition at line 1207 of file STM8L10x.h.

5.2.2.324 _I2C_TRISER_RESET_VALUE

```
#define _I2C_TRISER_RESET_VALUE ((uint8_t) 0x02)
```

I2C rise time register reset value.

Definition at line 1223 of file STM8L10x.h.

5.2.2.325 _I2C_TXE

```
#define _I2C_TXE ((uint8_t) (0x01 << 7))
```

I2C Data register empty (transmitters) [0] (in _I2C_SR1)

Definition at line 1276 of file STM8L10x.h.

5.2.2.326 _I2C_WUFH

```
#define _I2C_WUFH ((uint8_t) (0x01 << 5))
```

I2C Wakeup from Halt [0] (in _I2C_SR2)

Definition at line 1284 of file STM8L10x.h.

5.2.2.327 _IRTIM

```
#define _IRTIM _SFR(IRTIM_t, IRTIM_AddressBase)
```

IRTIM struct/bit access.

Definition at line 2357 of file STM8L10x.h.

5.2.2.328 _IRTIM_CR1

```
#define _IRTIM_CR1 _SFR(uint8_t, IRTIM_AddressBase+0x00)
```

IRTIM control register.

Definition at line 2358 of file STM8L10x.h.

5.2.2.329 _IRTIM_CR_RESET_VALUE

```
#define _IRTIM_CR_RESET_VALUE ((uint8_t) 0x00)
```

IRTIM control register reset value.

Definition at line 2361 of file STM8L10x.h.

5.2.2.330 _IRTIM_HS_EN

```
#define _IRTIM_HS_EN ((uint8_t) (0x01 << 1))
```

IRTIM High Sink LED driver enable [0] (in _IRTIM_CR)

Definition at line 2365 of file STM8L10x.h.

5.2.2.331 _IRTIM_IR_EN

```
#define _IRTIM_IR_EN ((uint8_t) (0x01 << 0))
```

IRTIM Infrared output enable [0] (in _IRTIM_CR)

Definition at line 2364 of file STM8L10x.h.

5.2.2.332 _ITC

```
#define _ITC _SFR(ITC_t, ITC_AddressBase)
```

ITC struct/bit access.

Definition at line 2572 of file STM8L10x.h.

5.2.2.333 _ITC_SPR1

```
#define _ITC_SPR1 _SFR(uint8_t, ITC_AddressBase+0x00)
```

Interrupt priority register 1/8.

Definition at line 2573 of file STM8L10x.h.

5.2.2.334 _ITC_SPR1_RESET_VALUE

```
#define _ITC_SPR1_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 1/8 reset value.

Definition at line 2583 of file STM8L10x.h.

5.2.2.335 _ITC_SPR2

```
#define _ITC_SPR2 _SFR(uint8_t, ITC_AddressBase+0x01)
```

Interrupt priority register 2/8.

Definition at line 2574 of file STM8L10x.h.

5.2.2.336 _ITC_SPR2_RESET_VALUE

```
#define _ITC_SPR2_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 2/8 reset value.

Definition at line 2584 of file STM8L10x.h.

5.2.2.337 _ITC_SPR3

```
#define _ITC_SPR3 _SFR(uint8_t, ITC_AddressBase+0x02)
```

Interrupt priority register 3/8.

Definition at line 2575 of file STM8L10x.h.

5.2.2.338 _ITC_SPR3_RESET_VALUE

```
#define _ITC_SPR3_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 3/8 reset value.

Definition at line 2585 of file STM8L10x.h.

5.2.2.339 _ITC_SPR4

```
#define _ITC_SPR4 _SFR(uint8_t, ITC_AddressBase+0x03)
```

Interrupt priority register 4/8.

Definition at line 2576 of file STM8L10x.h.

5.2.2.340 _ITC_SPR4_RESET_VALUE

```
#define _ITC_SPR4_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 4/8 reset value.

Definition at line 2586 of file STM8L10x.h.

5.2.2.341 _ITC_SPR5

```
#define _ITC_SPR5 _SFR(uint8_t, ITC_AddressBase+0x04)
```

Interrupt priority register 5/8.

Definition at line 2577 of file STM8L10x.h.

5.2.2.342 _ITC_SPR5_RESET_VALUE

```
#define _ITC_SPR5_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 5/8 reset value.

Definition at line 2587 of file STM8L10x.h.

5.2.2.343 _ITC_SPR6

```
#define _ITC_SPR6 _SFR(uint8_t, ITC_AddressBase+0x05)
```

Interrupt priority register 6/8.

Definition at line 2578 of file STM8L10x.h.

5.2.2.344 _ITC_SPR6_RESET_VALUE

```
#define _ITC_SPR6_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 6/8 reset value.

Definition at line 2588 of file STM8L10x.h.

5.2.2.345 _ITC_SPR7

```
#define _ITC_SPR7 _SFR(uint8_t, ITC_AddressBase+0x06)
```

Interrupt priority register 7/8.

Definition at line 2579 of file STM8L10x.h.

5.2.2.346 _ITC_SPR7_RESET_VALUE

```
#define _ITC_SPR7_RESET_VALUE ((uint8_t) 0xFF)
```

Interrupt priority register 7/8 reset value.

Definition at line 2589 of file STM8L10x.h.

5.2.2.347 _ITC_SPR8

```
#define _ITC_SPR8 _SFR(uint8_t, ITC_AddressBase+0x07)
```

Interrupt priority register 8/8.

Definition at line 2580 of file STM8L10x.h.

5.2.2.348 _ITC_SPR8_RESET_VALUE

```
#define _ITC_SPR8_RESET_VALUE ((uint8_t) 0x0F)
```

Interrupt priority register 8/8 reset value.

Definition at line 2590 of file STM8L10x.h.

5.2.2.349 _ITC_VECT10SPR

```
#define _ITC_VECT10SPR ((uint8_t) (0x03 << 4))
```

ITC interrupt priority vector 10 [1:0] (in _ITC_SPR3)

Definition at line 2618 of file STM8L10x.h.

5.2.2.350 _ITC_VECT10SPR0

```
#define _ITC_VECT10SPR0 ((uint8_t) (0x01 << 4))
```

ITC interrupt priority vector 10 [0] (in _ITC_SPR3)

Definition at line 2619 of file STM8L10x.h.

5.2.2.351 _ITC_VECT10SPR1

```
#define _ITC_VECT10SPR1 ((uint8_t) (0x01 << 5))
```

ITC interrupt priority vector 10 [1] (in _ITC_SPR3)

Definition at line 2620 of file STM8L10x.h.

5.2.2.352 _ITC_VECT11SPR

```
#define _ITC_VECT11SPR ((uint8_t) (0x03 << 6))
```

ITC interrupt priority vector 11 [1:0] (in _ITC_SPR3)

Definition at line 2621 of file STM8L10x.h.

5.2.2.353 _ITC_VECT11SPR0

```
#define _ITC_VECT11SPR0 ((uint8_t) (0x01 << 6))
```

ITC interrupt priority vector 11 [0] (in _ITC_SPR3)

Definition at line 2622 of file STM8L10x.h.

5.2.2.354 _ITC_VECT11SPR1

```
#define _ITC_VECT11SPR1 ((uint8_t) (0x01 << 7))
```

ITC interrupt priority vector 11 [1] (in _ITC_SPR3)

Definition at line 2623 of file STM8L10x.h.

5.2.2.355 _ITC_VECT12SPR

```
#define _ITC_VECT12SPR ((uint8_t) (0x03 << 0))
```

ITC interrupt priority vector 12 [1:0] (in _ITC_SPR4)

Definition at line 2626 of file STM8L10x.h.

5.2.2.356 _ITC_VECT12SPR0

```
#define _ITC_VECT12SPR0 ((uint8_t) (0x01 << 0))
```

ITC interrupt priority vector 12 [0] (in _ITC_SPR4)

Definition at line 2627 of file STM8L10x.h.

5.2.2.357 _ITC_VECT12SPR1

```
#define _ITC_VECT12SPR1 ((uint8_t) (0x01 << 1))
```

ITC interrupt priority vector 12 [1] (in _ITC_SPR4)

Definition at line 2628 of file STM8L10x.h.

5.2.2.358 _ITC_VECT13SPR

```
#define _ITC_VECT13SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 13 [1:0] (in _ITC_SPR4)

Definition at line 2629 of file STM8L10x.h.

5.2.2.359 _ITC_VECT13SPR0

```
#define _ITC_VECT13SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 13 [0] (in _ITC_SPR4)

Definition at line 2630 of file STM8L10x.h.

5.2.2.360 _ITC_VECT13SPR1

```
#define _ITC_VECT13SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 13 [1] (in _ITC_SPR4)

Definition at line 2631 of file STM8L10x.h.

5.2.2.361 _ITC_VECT14SPR

```
#define _ITC_VECT14SPR ((uint8_t) (0x03 << 4))
```

ITC interrupt priority vector 14 [1:0] (in _ITC_SPR4)

Definition at line 2632 of file STM8L10x.h.

5.2.2.362 _ITC_VECT14SPR0

```
#define _ITC_VECT14SPR0 ((uint8_t) (0x01 << 4))
```

ITC interrupt priority vector 14 [0] (in _ITC_SPR4)

Definition at line 2633 of file STM8L10x.h.

5.2.2.363 _ITC_VECT14SPR1

```
#define _ITC_VECT14SPR1 ((uint8_t) (0x01 << 5))
```

ITC interrupt priority vector 14 [1] (in _ITC_SPR4)

Definition at line 2634 of file STM8L10x.h.

5.2.2.364 _ITC_VECT15SPR

```
#define _ITC_VECT15SPR ((uint8_t) (0x03 << 6))
```

ITC interrupt priority vector 15 [1:0] (in _ITC_SPR4)

Definition at line 2635 of file STM8L10x.h.

5.2.2.365 _ITC_VECT15SPR0

```
#define _ITC_VECT15SPR0 ((uint8_t) (0x01 << 6))
```

ITC interrupt priority vector 15 [0] (in _ITC_SPR4)

Definition at line 2636 of file STM8L10x.h.

5.2.2.366 _ITC_VECT15SPR1

```
#define _ITC_VECT15SPR1 ((uint8_t) (0x01 << 7))
```

ITC interrupt priority vector 15 [1] (in _ITC_SPR4)

Definition at line 2637 of file STM8L10x.h.

5.2.2.367 _ITC_VECT19SPR

```
#define _ITC_VECT19SPR ((uint8_t) (0x03 << 6))
```

ITC interrupt priority vector 19 [1:0] (in _ITC_SPR5)

Definition at line 2641 of file STM8L10x.h.

5.2.2.368 _ITC_VECT19SPR0

```
#define _ITC_VECT19SPR0 ((uint8_t) (0x01 << 6))
```

ITC interrupt priority vector 19 [0] (in _ITC_SPR5)

Definition at line 2642 of file STM8L10x.h.

5.2.2.369 _ITC_VECT19SPR1

```
#define _ITC_VECT19SPR1 ((uint8_t) (0x01 << 7))
```

ITC interrupt priority vector 19 [1] (in _ITC_SPR5)

Definition at line 2643 of file STM8L10x.h.

5.2.2.370 _ITC_VECT1SPR

```
#define _ITC_VECT1SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 1 [1:0] (in _ITC_SPR1)

Definition at line 2594 of file STM8L10x.h.

5.2.2.371 _ITC_VECT1SPR0

```
#define _ITC_VECT1SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 1 [0] (in _ITC_SPR1)

Definition at line 2595 of file STM8L10x.h.

5.2.2.372 _ITC_VECT1SPR1

```
#define _ITC_VECT1SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 1 [1] (in _ITC_SPR1)

Definition at line 2596 of file STM8L10x.h.

5.2.2.373 _ITC_VECT20SPR

```
#define _ITC_VECT20SPR ((uint8_t) (0x03 << 0))
```

ITC interrupt priority vector 20 [1:0] (in _ITC_SPR6)

Definition at line 2646 of file STM8L10x.h.

5.2.2.374 _ITC_VECT20SPR0

```
#define _ITC_VECT20SPR0 ((uint8_t) (0x01 << 0))
```

ITC interrupt priority vector 20 [0] (in _ITC_SPR6)

Definition at line 2647 of file STM8L10x.h.

5.2.2.375 _ITC_VECT20SPR1

```
#define _ITC_VECT20SPR1 ((uint8_t) (0x01 << 1))
```

ITC interrupt priority vector 20 [1] (in _ITC_SPR6)

Definition at line 2648 of file STM8L10x.h.

5.2.2.376 _ITC_VECT21SPR

```
#define _ITC_VECT21SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 21 [1:0] (in _ITC_SPR6)

Definition at line 2649 of file STM8L10x.h.

5.2.2.377 _ITC_VECT21SPR0

```
#define _ITC_VECT21SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 21 [0] (in _ITC_SPR6)

Definition at line 2650 of file STM8L10x.h.

5.2.2.378 _ITC_VECT21SPR1

```
#define _ITC_VECT21SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 21 [1] (in _ITC_SPR6)

Definition at line 2651 of file STM8L10x.h.

5.2.2.379 _ITC_VECT22SPR

```
#define _ITC_VECT22SPR ((uint8_t) (0x03 << 4))
```

ITC interrupt priority vector 22 [1:0] (in _ITC_SPR6)

Definition at line 2652 of file STM8L10x.h.

5.2.2.380 _ITC_VECT22SPR0

```
#define _ITC_VECT22SPR0 ((uint8_t) (0x01 << 4))
```

ITC interrupt priority vector 22 [0] (in _ITC_SPR6)

Definition at line 2653 of file STM8L10x.h.

5.2.2.381 _ITC_VECT22SPR1

```
#define _ITC_VECT22SPR1 ((uint8_t) (0x01 << 5))
```

ITC interrupt priority vector 22 [1] (in _ITC_SPR6)

Definition at line 2654 of file STM8L10x.h.

5.2.2.382 _ITC_VECT25SPR

```
#define _ITC_VECT25SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 25 [1:0] (in _ITC_SPR7)

Definition at line 2659 of file STM8L10x.h.

5.2.2.383 _ITC_VECT25SPR0

```
#define _ITC_VECT25SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 25 [0] (in _ITC_SPR7)

Definition at line 2660 of file STM8L10x.h.

5.2.2.384 _ITC_VECT25SPR1

```
#define _ITC_VECT25SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 25 [1] (in _ITC_SPR7)

Definition at line 2661 of file STM8L10x.h.

5.2.2.385 _ITC_VECT26SPR

```
#define _ITC_VECT26SPR ((uint8_t) (0x03 << 4))
```

ITC interrupt priority vector 26 [1:0] (in _ITC_SPR7)

Definition at line 2662 of file STM8L10x.h.

5.2.2.386 _ITC_VECT26SPR0

```
#define _ITC_VECT26SPR0 ((uint8_t) (0x01 << 4))
```

ITC interrupt priority vector 26 [0] (in _ITC_SPR7)

Definition at line 2663 of file STM8L10x.h.

5.2.2.387 _ITC_VECT26SPR1

```
#define _ITC_VECT26SPR1 ((uint8_t) (0x01 << 5))
```

ITC interrupt priority vector 26 [1] (in _ITC_SPR7)

Definition at line 2664 of file STM8L10x.h.

5.2.2.388 _ITC_VECT27SPR

```
#define _ITC_VECT27SPR ((uint8_t) (0x03 << 6))
```

ITC interrupt priority vector 27 [1:0] (in _ITC_SPR7)

Definition at line 2665 of file STM8L10x.h.

5.2.2.389 _ITC_VECT27SPR0

```
#define _ITC_VECT27SPR0 ((uint8_t) (0x01 << 6))
```

ITC interrupt priority vector 27 [0] (in _ITC_SPR7)

Definition at line 2666 of file STM8L10x.h.

5.2.2.390 _ITC_VECT27SPR1

```
#define _ITC_VECT27SPR1 ((uint8_t) (0x01 << 7))
```

ITC interrupt priority vector 27 [1] (in _ITC_SPR7)

Definition at line 2667 of file STM8L10x.h.

5.2.2.391 _ITC_VECT28SPR

```
#define _ITC_VECT28SPR ((uint8_t) (0x03 << 0))
```

ITC interrupt priority vector 28 [1:0] (in _ITC_SPR8)

Definition at line 2670 of file STM8L10x.h.

5.2.2.392 _ITC_VECT28SPR0

```
#define _ITC_VECT28SPR0 ((uint8_t) (0x01 << 0))
```

ITC interrupt priority vector 28 [0] (in _ITC_SPR8)

Definition at line 2671 of file STM8L10x.h.

5.2.2.393 _ITC_VECT28SPR1

```
#define _ITC_VECT28SPR1 ((uint8_t) (0x01 << 1))
```

ITC interrupt priority vector 28 [1] (in _ITC_SPR8)

Definition at line 2672 of file STM8L10x.h.

5.2.2.394 _ITC_VECT29SPR

```
#define _ITC_VECT29SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 29 [1:0] (in _ITC_SPR8)

Definition at line 2673 of file STM8L10x.h.

5.2.2.395 _ITC_VECT29SPR0

```
#define _ITC_VECT29SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 29 [0] (in _ITC_SPR8)

Definition at line 2674 of file STM8L10x.h.

5.2.2.396 _ITC_VECT29SPR1

```
#define _ITC_VECT29SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 29 [1] (in _ITC_SPR8)

Definition at line 2675 of file STM8L10x.h.

5.2.2.397 _ITC_VECT4SPR

```
#define _ITC_VECT4SPR ((uint8_t) (0x03 << 0))
```

ITC interrupt priority vector 4 [1:0] (in _ITC_SPR2)

Definition at line 2600 of file STM8L10x.h.

5.2.2.398 _ITC_VECT4SPR0

```
#define _ITC_VECT4SPR0 ((uint8_t) (0x01 << 0))
```

ITC interrupt priority vector 4 [0] (in _ITC_SPR2)

Definition at line 2601 of file STM8L10x.h.

5.2.2.399 _ITC_VECT4SPR1

```
#define _ITC_VECT4SPR1 ((uint8_t) (0x01 << 1))
```

ITC interrupt priority vector 4 [1] (in _ITC_SPR2)

Definition at line 2602 of file STM8L10x.h.

5.2.2.400 _ITC_VECT6SPR

```
#define _ITC_VECT6SPR ((uint8_t) (0x03 << 4))
```

ITC interrupt priority vector 6 [1:0] (in _ITC_SPR2)

Definition at line 2604 of file STM8L10x.h.

5.2.2.401 _ITC_VECT6SPR0

```
#define _ITC_VECT6SPR0 ((uint8_t) (0x01 << 4))
```

ITC interrupt priority vector 6 [0] (in _ITC_SPR2)

Definition at line 2605 of file STM8L10x.h.

5.2.2.402 _ITC_VECT6SPR1

```
#define _ITC_VECT6SPR1 ((uint8_t) (0x01 << 5))
```

ITC interrupt priority vector 6 [1] (in _ITC_SPR2)

Definition at line 2606 of file STM8L10x.h.

5.2.2.403 _ITC_VECT7SPR

```
#define _ITC_VECT7SPR ((uint8_t) (0x03 << 6))
```

ITC interrupt priority vector 7 [1:0] (in _ITC_SPR2)

Definition at line 2607 of file STM8L10x.h.

5.2.2.404 _ITC_VECT7SPR0

```
#define _ITC_VECT7SPR0 ((uint8_t) (0x01 << 6))
```

ITC interrupt priority vector 7 [0] (in _ITC_SPR2)

Definition at line 2608 of file STM8L10x.h.

5.2.2.405 _ITC_VECT7SPR1

```
#define _ITC_VECT7SPR1 ((uint8_t) (0x01 << 7))
```

ITC interrupt priority vector 7 [1] (in _ITC_SPR2)

Definition at line 2609 of file STM8L10x.h.

5.2.2.406 _ITC_VECT8SPR

```
#define _ITC_VECT8SPR ((uint8_t) (0x03 << 0))
```

ITC interrupt priority vector 8 [1:0] (in _ITC_SPR3)

Definition at line 2612 of file STM8L10x.h.

5.2.2.407 _ITC_VECT8SPR0

```
#define _ITC_VECT8SPR0 ((uint8_t) (0x01 << 0))
```

ITC interrupt priority vector 8 [0] (in _ITC_SPR3)

Definition at line 2613 of file STM8L10x.h.

5.2.2.408 _ITC_VECT8SPR1

```
#define _ITC_VECT8SPR1 ((uint8_t) (0x01 << 1))
```

ITC interrupt priority vector 8 [1] (in _ITC_SPR3)

Definition at line 2614 of file STM8L10x.h.

5.2.2.409 _ITC_VECT9SPR

```
#define _ITC_VECT9SPR ((uint8_t) (0x03 << 2))
```

ITC interrupt priority vector 9 [1:0] (in _ITC_SPR3)

Definition at line 2615 of file STM8L10x.h.

5.2.2.410 _ITC_VECT9SPR0

```
#define _ITC_VECT9SPR0 ((uint8_t) (0x01 << 2))
```

ITC interrupt priority vector 9 [0] (in _ITC_SPR3)

Definition at line 2616 of file STM8L10x.h.

5.2.2.411 _ITC_VECT9SPR1

```
#define _ITC_VECT9SPR1 ((uint8_t) (0x01 << 3))
```

ITC interrupt priority vector 9 [1] (in _ITC_SPR3)

Definition at line 2617 of file STM8L10x.h.

5.2.2.412 _IWDG

```
#define _IWDG _SFR(IWDG_t, IWDG_AddressBase)
```

Independent Timeout Watchdog struct/bit access.

Definition at line 808 of file STM8L10x.h.

5.2.2.413 _IWDG_KEY_ACCESS

```
#define _IWDG_KEY_ACCESS ((uint8_t) 0x55)
```

Independent Timeout Watchdog unlock write to IWDG_PR and IWDG_RLR (in _IWDG_KR)

Definition at line 820 of file STM8L10x.h.

5.2.2.414 _IWDG_KEY_ENABLE

```
#define _IWDG_KEY_ENABLE ((uint8_t) 0xCC)
```

Independent Timeout Watchdog enable (in _IWDG_KR)

Definition at line 818 of file STM8L10x.h.

5.2.2.415 _IWDG_KEY_REFRESH

```
#define _IWDG_KEY_REFRESH ((uint8_t) 0xAA)
```

Independent Timeout Watchdog refresh (in _IWDG_KR)

Definition at line 819 of file STM8L10x.h.

5.2.2.416 _IWDG_KR

```
#define _IWDG_KR _SFR(uint8_t, IWDG_AddressBase+0x00)
```

Independent Timeout Watchdog Key register (IWDG_KR)

Definition at line 809 of file STM8L10x.h.

5.2.2.417 _IWDG_PR

```
#define _IWDG_PR _SFR(uint8_t, IWDG_AddressBase+0x01)
```

Independent Timeout Watchdog Prescaler register (IWDG_PR)

Definition at line 810 of file STM8L10x.h.

5.2.2.418 _IWDG_PR_RESET_VALUE

```
#define _IWDG_PR_RESET_VALUE ((uint8_t) 0x00)
```

Independent Timeout Watchdog Prescaler register reset value.

Definition at line 814 of file STM8L10x.h.

5.2.2.419 _IWDG_PRE

```
#define _IWDG_PRE ((uint8_t) (0x07 << 0))
```

Independent Timeout Watchdog Prescaler divider [2:0] (in _IWDG_PR)

Definition at line 823 of file STM8L10x.h.

5.2.2.420 _IWDG_PRE0

```
#define _IWDG_PRE0 ((uint8_t) (0x01 << 0))
```

Independent Timeout Watchdog Prescaler divider [0] (in _IWDG_PR)

Definition at line 824 of file STM8L10x.h.

5.2.2.421 _IWDG_PRE1

```
#define _IWDG_PRE1 ((uint8_t) (0x01 << 1))
```

Independent Timeout Watchdog Prescaler divider [1] (in _IWDG_PR)

Definition at line 825 of file STM8L10x.h.

5.2.2.422 _IWDG_PRE2

```
#define _IWDG_PRE2 ((uint8_t) (0x01 << 2))
```

Independent Timeout Watchdog Prescaler divider [2] (in _IWDG_PR)

Definition at line 826 of file STM8L10x.h.

5.2.2.423 _IWDG_RLR

```
#define _IWDG_RLR _SFR(uint8_t, IWDG_AddressBase+0x02)
```

Independent Timeout Watchdog Reload register (IWDG_RLR)

Definition at line 811 of file STM8L10x.h.

5.2.2.424 _IWDG_RLR_RESET_VALUE

```
#define _IWDG_RLR_RESET_VALUE ((uint8_t) 0xFF)
```

Independent Timeout Watchdog Reload register reset value.

Definition at line 815 of file STM8L10x.h.

5.2.2.425 _RST

```
#define _RST _SFR(RST_t, RST_AddressBase)
```

Reset module struct/bit access.

Definition at line 669 of file STM8L10x.h.

5.2.2.426 _RST_CR

```
#define _RST_CR _SFR(uint8_t, RST_AddressBase+0x00)
```

Reset pin configuration register (RST_CR)

Definition at line 670 of file STM8L10x.h.

5.2.2.427 _RST_ILLOPF

```
#define _RST_ILLOPF ((uint8_t) (0x01 << 2))
```

Illegal opcode reset flag [0] (in _RST_SR)

Definition at line 679 of file STM8L10x.h.

5.2.2.428 _RST_IWDGF

```
#define _RST_IWDGF ((uint8_t) (0x01 << 1))
```

Independent Watchdog reset flag [0] (in _RST_SR)

Definition at line 678 of file STM8L10x.h.

5.2.2.429 _RST_PIN_KEY

```
#define _RST_PIN_KEY ((uint8_t) 0xD0)
```

Configure PA1 as GPIO, else NRST (in _RST_CR)

Definition at line 674 of file STM8L10x.h.

5.2.2.430 _RST_PORF

```
#define _RST_PORF ((uint8_t) (0x01 << 0))
```

Power-on reset (POR) flag [0] (in _RST_SR)

Definition at line 677 of file STM8L10x.h.

5.2.2.431 _RST_SR

```
#define _RST_SR _SFR(uint8_t, RST_AddressBase+0x01)
```

Reset module status register (RST_SR)

Definition at line 671 of file STM8L10x.h.

5.2.2.432 _RST_SWIMF

```
#define _RST_SWIMF ((uint8_t) (0x01 << 3))
```

SWIM reset flag [0] (in _RST_SR)

Definition at line 680 of file STM8L10x.h.

5.2.2.433 _SFR

```
#define _SFR(
              type,
              addr ) (*((volatile type*) (addr)))
```

peripheral register

Definition at line 187 of file STM8L10x.h.

5.2.2.434 _SPI

```
#define _SPI _SFR(SPI_t, SPI_AddressBase)
```

register for SPI control

SPI struct/bit access

Definition at line 1007 of file STM8L10x.h.

5.2.2.435 _SPI_BDM

```
#define _SPI_BDM ((uint8_t) (0x01 << 7))
```

SPI Bidirectional data mode enable [0] (in _SPI_CR2)

Definition at line 1038 of file STM8L10x.h.

5.2.2.436 _SPI_BDOE

```
#define _SPI_BDOE ((uint8_t) (0x01 << 6))
```

SPI Input/Output enable in bidirectional mode [0] (in _SPI_CR2)

Definition at line 1037 of file STM8L10x.h.

5.2.2.437 _SPI_BR

```
#define _SPI_BR ((uint8_t) (0x07 << 3))
```

SPI Baudrate control [2:0] (in _SPI_CR1)

Definition at line 1025 of file STM8L10x.h.

5.2.2.438 _SPI_BR0

```
#define _SPI_BR0 ((uint8_t) (0x01 << 3))
```

SPI Baudrate control [0] (in _SPI_CR1)

Definition at line 1026 of file STM8L10x.h.

5.2.2.439 _SPI_BR1

```
#define _SPI_BR1 ((uint8_t) (0x01 << 4))
```

SPI Baudrate control [1] (in _SPI_CR1)

Definition at line 1027 of file STM8L10x.h.

5.2.2.440 _SPI_BR2

```
#define _SPI_BR2 ((uint8_t) (0x01 << 5))
```

SPI Baudrate control [2] (in _SPI_CR1)

Definition at line 1028 of file STM8L10x.h.

5.2.2.441 _SPI_BSY

```
#define _SPI_BSY ((uint8_t) (0x01 << 7))
```

SPI Busy flag [0] (in _SPI_SR)

Definition at line 1055 of file STM8L10x.h.

5.2.2.442 _SPI_CPHA

```
#define _SPI_CPHA ((uint8_t) (0x01 << 0))
```

SPI Clock phase [0] (in _SPI_CR1)

Definition at line 1022 of file STM8L10x.h.

5.2.2.443 _SPI_CPOL

```
#define _SPI_CPOL ((uint8_t) (0x01 << 1))
```

SPI Clock polarity [0] (in _SPI_CR1)

Definition at line 1023 of file STM8L10x.h.

5.2.2.444 _SPI_CR1

```
#define _SPI_CR1 _SFR(uint8_t, SPI_AddressBase+0x00)
```

SPI control register 1.

Definition at line 1008 of file STM8L10x.h.

5.2.2.445 _SPI_CR1_RESET_VALUE

```
#define _SPI_CR1_RESET_VALUE ((uint8_t) 0x00)
```

SPI Control Register 1 reset value.

Definition at line 1015 of file STM8L10x.h.

5.2.2.446 _SPI_CR2

```
#define _SPI_CR2 _SFR(uint8_t, SPI_AddressBase+0x01)
```

SPI control register 2.

Definition at line 1009 of file STM8L10x.h.

5.2.2.447 _SPI_CR2_RESET_VALUE

```
#define _SPI_CR2_RESET_VALUE ((uint8_t) 0x00)
```

SPI Control Register 2 reset value.

Definition at line 1016 of file STM8L10x.h.

5.2.2.448 _SPI_DR

```
#define _SPI_DR _SFR(uint8_t, SPI_AddressBase+0x04)
```

SPI data register.

Definition at line 1012 of file STM8L10x.h.

5.2.2.449 _SPI_DR_RESET_VALUE

```
#define _SPI_DR_RESET_VALUE ((uint8_t) 0x00)
```

SPI Data Register reset value.

Definition at line 1019 of file STM8L10x.h.

5.2.2.450 _SPI_ERRIE

```
#define _SPI_ERRIE ((uint8_t) (0x01 << 5))
```

SPI Error interrupt enable [0] (in _SPI_ICR)

Definition at line 1043 of file STM8L10x.h.

5.2.2.451 _SPI_ICR

```
#define _SPI_ICR _SFR(uint8_t, SPI_AddressBase+0x02)
```

SPI interrupt control register.

Definition at line 1010 of file STM8L10x.h.

5.2.2.452 _SPI_ICR_RESET_VALUE

```
#define _SPI_ICR_RESET_VALUE ((uint8_t) 0x00)
```

SPI Interrupt Control Register reset value.

Definition at line 1017 of file STM8L10x.h.

5.2.2.453 _SPI_LSBFIRST

```
#define _SPI_LSBFIRST ((uint8_t) (0x01 << 7))
```

SPI Frame format [0] (in _SPI_CR1)

Definition at line 1030 of file STM8L10x.h.

5.2.2.454 _SPI_MODF

```
#define _SPI_MODF ((uint8_t) (0x01 << 5))
```

SPI Mode fault [0] (in _SPI_SR)

Definition at line 1053 of file STM8L10x.h.

5.2.2.455 _SPI_MSTR

```
#define _SPI_MSTR ((uint8_t) (0x01 << 2))
```

SPI Master/slave selection [0] (in _SPI_CR1)

Definition at line 1024 of file STM8L10x.h.

5.2.2.456 _SPI_OVR

```
#define _SPI_OVR ((uint8_t) (0x01 << 6))
```

SPI Overrun flag [0] (in _SPI_SR)

Definition at line 1054 of file STM8L10x.h.

5.2.2.457 _SPI_RXIE

```
#define _SPI_RXIE ((uint8_t) (0x01 << 6))
```

SPI Rx buffer not empty interrupt enable [0] (in _SPI_ICR)

Definition at line 1044 of file STM8L10x.h.

5.2.2.458 _SPI_RXNE

```
#define _SPI_RXNE ((uint8_t) (0x01 << 0))
```

SPI Receive buffer not empty [0] (in _SPI_SR)

Definition at line 1048 of file STM8L10x.h.

5.2.2.459 _SPI_RXONLY

```
#define _SPI_RXONLY ((uint8_t) (0x01 << 2))
```

SPI Receive only [0] (in _SPI_CR2)

Definition at line 1035 of file STM8L10x.h.

5.2.2.460 _SPI_SPE

```
#define _SPI_SPE ((uint8_t) (0x01 << 6))
```

SPI enable [0] (in _SPI_CR1)

Definition at line 1029 of file STM8L10x.h.

5.2.2.461 _SPI_SR

```
#define _SPI_SR _SFR(uint8_t, SPI_AddressBase+0x03)
```

SPI status register.

Definition at line 1011 of file STM8L10x.h.

5.2.2.462 _SPI_SR_RESET_VALUE

```
#define _SPI_SR_RESET_VALUE ((uint8_t) 0x02)
```

SPI Status Register reset value.

Definition at line 1018 of file STM8L10x.h.

5.2.2.463 _SPI_SSI

```
#define _SPI_SSI ((uint8_t) (0x01 << 0))
```

SPI Internal slave select [0] (in _SPI_CR2)

Definition at line 1033 of file STM8L10x.h.

5.2.2.464 _SPI_SSM

```
#define _SPI_SSM ((uint8_t) (0x01 << 1))
```

SPI Software slave management [0] (in _SPI_CR2)

Definition at line 1034 of file STM8L10x.h.

5.2.2.465 _SPI_TXE

```
#define _SPI_TXE ((uint8_t) (0x01 << 1))
```

SPI Transmit buffer empty [0] (in _SPI_SR)

Definition at line 1049 of file STM8L10x.h.

5.2.2.466 _SPI_TXIE

```
#define _SPI_TXIE ((uint8_t) (0x01 << 7))
```

SPI Tx buffer empty interrupt enable [0] (in _SPI_ICR)

Definition at line 1045 of file STM8L10x.h.

5.2.2.467 _SPI_WKIE

```
#define _SPI_WKIE ((uint8_t) (0x01 << 4))
```

SPI Wakeup interrupt enable [0] (in _SPI_ICR)

Definition at line 1042 of file STM8L10x.h.

5.2.2.468 _SPI_WKUP

```
#define _SPI_WKUP ((uint8_t) (0x01 << 3))
```

SPI Wakeup flag [0] (in _SPI_SR)

Definition at line 1051 of file STM8L10x.h.

5.2.2.469 _TIM2

```
#define _TIM2 _SFR(TIM2_3_t, TIM2_AddressBase)
```

TIM2 struct/bit access.

Definition at line 1769 of file STM8L10x.h.

5.2.2.470 _TIM2_AOE

```
#define _TIM2_AOE ((uint8_t) (0x01 << 6))
```

TIM2 Automatic output enable [0] (in _TIM2_BKR)

Definition at line 1958 of file STM8L10x.h.

5.2.2.471 _TIM2_ARPE

```
#define _TIM2_ARPE ((uint8_t) (0x01 << 7))
```

TIM2 Auto-reload preload enable [0] (in _TIM2_CR1)

Definition at line 1825 of file STM8L10x.h.

5.2.2.472 _TIM2_ARRH

```
#define _TIM2_ARRH _SFR(uint8_t, TIM2_AddressBase+0x0E)
```

TIM2 auto-reload register high byte.

Definition at line 1784 of file STM8L10x.h.

5.2.2.473 _TIM2_ARRH_RESET_VALUE

```
#define _TIM2_ARRH_RESET_VALUE ((uint8_t) 0xFF)
```

TIM2 auto-reload register high byte reset value.

Definition at line 1808 of file STM8L10x.h.

5.2.2.474 _TIM2_ARRL

```
#define _TIM2_ARRL _SFR(uint8_t, TIM2_AddressBase+0x0F)
```

TIM2 auto-reload register low byte.

Definition at line 1785 of file STM8L10x.h.

5.2.2.475 _TIM2_ARRL_RESET_VALUE

```
#define _TIM2_ARRL_RESET_VALUE ((uint8_t) 0xFF)
```

TIM2 auto-reload register low byte reset value.

Definition at line 1809 of file STM8L10x.h.

5.2.2.476 _TIM2_BG

```
#define _TIM2_BG ((uint8_t) (0x01 << 7))
```

TIM2 Break generation [0] (in _TIM2_EGR)

Definition at line 1887 of file STM8L10x.h.

5.2.2.477 _TIM2_BIE

```
#define _TIM2_BIE ((uint8_t) (0x01 << 7))
```

TIM2 Break interrupt enable [0] (in _TIM2_IER)

Definition at line 1865 of file STM8L10x.h.

5.2.2.478 _TIM2_BIF

```
#define _TIM2_BIF ((uint8_t) (0x01 << 7))
```

TIM2 Break interrupt flag [0] (in _TIM2_SR1)

Definition at line 1873 of file STM8L10x.h.

5.2.2.479 _TIM2_BKE

```
#define _TIM2_BKE ((uint8_t) (0x01 << 4))
```

TIM2 Break enable [0] (in _TIM2_BKR)

Definition at line 1956 of file STM8L10x.h.

5.2.2.480 _TIM2_BKP

```
#define _TIM2_BKP ((uint8_t) (0x01 << 5))
```

TIM2 Break polarity [0] (in _TIM2_BKR)

Definition at line 1957 of file STM8L10x.h.

5.2.2.481 _TIM2_BKR

```
#define _TIM2_BKR _SFR(uint8_t, TIM2_AddressBase+0x14)
```

TIM2 Break register.

Definition at line 1790 of file STM8L10x.h.

5.2.2.482 _TIM2_BKR_RESET_VALUE

```
#define _TIM2_BKR_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Break register reset value.

Definition at line 1815 of file STM8L10x.h.

5.2.2.483 _TIM2_CC1E

```
#define _TIM2_CC1E ((uint8_t) (0x01 << 0))
```

TIM2 Capture/compare 1 output enable [0] (in _TIM2_CCER1)

Definition at line 1936 of file STM8L10x.h.

5.2.2.484 _TIM2_CC1G

```
#define _TIM2_CC1G ((uint8_t) (0x01 << 1))
```

TIM2 Capture/compare 1 generation [0] (in _TIM2_EGR)

Definition at line 1883 of file STM8L10x.h.

5.2.2.485 _TIM2_CC1IE

```
#define _TIM2_CC1IE ((uint8_t) (0x01 << 1))
```

TIM2 Capture/compare 1 interrupt enable [0] (in _TIM2_IER)

Definition at line 1861 of file STM8L10x.h.

5.2.2.486 _TIM2_CC1IF

```
#define _TIM2_CC1IF ((uint8_t) (0x01 << 1))
```

TIM2 Capture/compare 1 interrupt flag [0] (in _TIM2_SR1)

Definition at line 1869 of file STM8L10x.h.

5.2.2.487 _TIM2_CC1OF

```
#define _TIM2_CC1OF ((uint8_t) (0x01 << 1))
```

TIM2 Capture/compare 1 overcapture flag [0] (in _TIM2_SR2)

Definition at line 1877 of file STM8L10x.h.

5.2.2.488 _TIM2_CC1P

```
#define _TIM2_CC1P ((uint8_t) (0x01 << 1))
```

TIM2 Capture/compare 1 output polarity [0] (in _TIM2_CCER1)

Definition at line 1937 of file STM8L10x.h.

5.2.2.489 _TIM2_CC1S

```
#define _TIM2_CC1S ((uint8_t) (0x03 << 0))
```

TIM2 Compare 1 selection [1:0] (in _TIM2_CCMR1)

Definition at line 1890 of file STM8L10x.h.

5.2.2.490 _TIM2_CC1S0

```
#define _TIM2_CC1S0 ((uint8_t) (0x01 << 0))
```

TIM2 Compare 1 selection [0] (in _TIM2_CCMR1)

Definition at line 1891 of file STM8L10x.h.

5.2.2.491 _TIM2_CC1S1

```
#define _TIM2_CC1S1 ((uint8_t) (0x01 << 1))
```

TIM2 Compare 1 selection [1] (in _TIM2_CCMR1)

Definition at line 1892 of file STM8L10x.h.

5.2.2.492 _TIM2_CC2E

```
#define _TIM2_CC2E ((uint8_t) (0x01 << 4))
```

TIM2 Capture/compare 2 output enable [0] (in _TIM2_CCER1)

Definition at line 1939 of file STM8L10x.h.

5.2.2.493 _TIM2_CC2G

```
#define _TIM2_CC2G ((uint8_t) (0x01 << 2))
```

TIM2 Capture/compare 2 generation [0] (in _TIM2_EGR)

Definition at line 1884 of file STM8L10x.h.

5.2.2.494 _TIM2_CC2IE

```
#define _TIM2_CC2IE ((uint8_t) (0x01 << 2))
```

TIM2 Capture/compare 2 interrupt enable [0] (in _TIM2_IER)

Definition at line 1862 of file STM8L10x.h.

5.2.2.495 _TIM2_CC2IF

```
#define _TIM2_CC2IF ((uint8_t) (0x01 << 2))
```

TIM2 Capture/compare 2 interrupt flag [0] (in _TIM2_SR1)

Definition at line 1870 of file STM8L10x.h.

5.2.2.496 _TIM2_CC2OF

```
#define _TIM2_CC2OF ((uint8_t) (0x01 << 2))
```

TIM2 Capture/compare 2 overcapture flag [0] (in _TIM2_SR2)

Definition at line 1878 of file STM8L10x.h.

5.2.2.497 _TIM2_CC2P

```
#define _TIM2_CC2P ((uint8_t) (0x01 << 5))
```

TIM2 Capture/compare 2 output polarity [0] (in _TIM2_CCER1)

Definition at line 1940 of file STM8L10x.h.

5.2.2.498 _TIM2_CC2S

```
#define _TIM2_CC2S ((uint8_t) (0x03 << 0))
```

TIM2 Compare 2 selection [1:0] (in _TIM2_CCMR2)

Definition at line 1913 of file STM8L10x.h.

5.2.2.499 _TIM2_CC2S0

```
#define _TIM2_CC2S0 ((uint8_t) (0x01 << 0))
```

TIM2 Compare 2 selection [0] (in _TIM2_CCMR2)

Definition at line 1914 of file STM8L10x.h.

5.2.2.500 _TIM2_CC2S1

```
#define _TIM2_CC2S1 ((uint8_t) (0x01 << 1))
```

TIM2 Compare 2 selection [1] (in _TIM2_CCMR2)

Definition at line 1915 of file STM8L10x.h.

5.2.2.501 _TIM2_CCER1

```
#define _TIM2_CCER1 _SFR(uint8_t, TIM2_AddressBase+0x0A)
```

TIM2 Capture/compare enable register 1.

Definition at line 1780 of file STM8L10x.h.

5.2.2.502 _TIM2_CCER1_RESET_VALUE

```
#define _TIM2_CCER1_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Capture/compare enable register 1 reset value.

Definition at line 1804 of file STM8L10x.h.

5.2.2.503 _TIM2_CCMR1

```
#define _TIM2_CCMR1 _SFR(uint8_t, TIM2_AddressBase+0x08)
```

TIM2 Capture/compare mode register 1.

Definition at line 1778 of file STM8L10x.h.

5.2.2.504 _TIM2_CCMR1_RESET_VALUE

```
#define _TIM2_CCMR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Capture/compare mode register 1 reset value.

Definition at line 1802 of file STM8L10x.h.

5.2.2.505 _TIM2_CCMR2

```
#define _TIM2_CCMR2 _SFR(uint8_t, TIM2_AddressBase+0x09)
```

TIM2 Capture/compare mode register 2.

Definition at line 1779 of file STM8L10x.h.

5.2.2.506 _TIM2_CCMR2_RESET_VALUE

```
#define _TIM2_CCMR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Capture/compare mode register 2 reset value.

Definition at line 1803 of file STM8L10x.h.

5.2.2.507 _TIM2_CCR1H

```
#define _TIM2_CCR1H _SFR(uint8_t, TIM2_AddressBase+0x10)
```

TIM2 16-bit capture/compare value 1 high byte.

Definition at line 1786 of file STM8L10x.h.

5.2.2.508 _TIM2_CCR1H_RESET_VALUE

```
#define _TIM2_CCR1H_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 16-bit capture/compare value 1 high byte reset value.

Definition at line 1811 of file STM8L10x.h.

5.2.2.509 _TIM2_CCR1L

```
#define _TIM2_CCR1L _SFR(uint8_t, TIM2_AddressBase+0x11)
```

TIM2 16-bit capture/compare value 1 low byte.

Definition at line 1787 of file STM8L10x.h.

5.2.2.510 _TIM2_CCR1L_RESET_VALUE

```
#define _TIM2_CCR1L_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 16-bit capture/compare value 1 low byte reset value.

Definition at line 1812 of file STM8L10x.h.

5.2.2.511 _TIM2_CCR2H

```
#define _TIM2_CCR2H _SFR(uint8_t, TIM2_AddressBase+0x12)
```

TIM2 16-bit capture/compare value 2 high byte.

Definition at line 1788 of file STM8L10x.h.

5.2.2.512 _TIM2_CCR2H_RESET_VALUE

```
#define _TIM2_CCR2H_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 16-bit capture/compare value 2 high byte reset value.

Definition at line 1813 of file STM8L10x.h.

5.2.2.513 _TIM2_CCR2L

```
#define _TIM2_CCR2L _SFR(uint8_t, TIM2_AddressBase+0x13)
```

TIM2 16-bit capture/compare value 2 low byte.

Definition at line 1789 of file STM8L10x.h.

5.2.2.514 _TIM2_CCR2L_RESET_VALUE

```
#define _TIM2_CCR2L_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 16-bit capture/compare value 2 low byte reset value.

Definition at line 1814 of file STM8L10x.h.

5.2.2.515 _TIM2_CEN

```
#define _TIM2_CEN ((uint8_t) (0x01 << 0))
```

TIM2 Counter enable [0] (in _TIM2_CR1)

Definition at line 1819 of file STM8L10x.h.

5.2.2.516 _TIM2_CNTRH

```
#define _TIM2_CNTRH _SFR(uint8_t, TIM2_AddressBase+0x0B)
```

TIM2 counter register high byte.

Definition at line 1781 of file STM8L10x.h.

5.2.2.517 _TIM2_CNTRH_RESET_VALUE

```
#define _TIM2_CNTRH_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 counter register high byte reset value.

Definition at line 1805 of file STM8L10x.h.

5.2.2.518 _TIM2_CNTRL

```
#define _TIM2_CNTRL _SFR(uint8_t, TIM2_AddressBase+0x0C)
```

TIM2 counter register low byte.

Definition at line 1782 of file STM8L10x.h.

5.2.2.519 _TIM2_CNTRL_RESET_VALUE

```
#define _TIM2_CNTRL_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 counter register low byte reset value.

Definition at line 1806 of file STM8L10x.h.

5.2.2.520 _TIM2_CR1

```
#define _TIM2_CR1 _SFR(uint8_t, TIM2_AddressBase+0x00)
```

TIM2 control register 1.

Definition at line 1770 of file STM8L10x.h.

5.2.2.521 _TIM2_CR1_RESET_VALUE

```
#define _TIM2_CR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 control register 1 reset value.

Definition at line 1794 of file STM8L10x.h.

5.2.2.522 _TIM2_CR2

```
#define _TIM2_CR2 _SFR(uint8_t, TIM2_AddressBase+0x01)
```

TIM2 control register 2.

Definition at line 1771 of file STM8L10x.h.

5.2.2.523 _TIM2_CR2_RESET_VALUE

```
#define _TIM2_CR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 control register 2 reset value.

Definition at line 1795 of file STM8L10x.h.

5.2.2.524 _TIM2_DIR

```
#define _TIM2_DIR ((uint8_t) (0x01 << 4))
```

TIM2 Direction [0] (in _TIM2_CR1)

Definition at line 1823 of file STM8L10x.h.

5.2.2.525 _TIM2_ECE

```
#define _TIM2_ECE ((uint8_t) (0x01 << 6))
```

TIM2 External clock enable [0] (in _TIM2_ETR)

Definition at line 1856 of file STM8L10x.h.

5.2.2.526 _TIM2_EGR

```
#define _TIM2_EGR _SFR(uint8_t, TIM2_AddressBase+0x07)
```

TIM2 Event generation register.

Definition at line 1777 of file STM8L10x.h.

5.2.2.527 _TIM2_EGR_RESET_VALUE

```
#define _TIM2_EGR_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Event generation register reset value.

Definition at line 1801 of file STM8L10x.h.

5.2.2.528 _TIM2_ETF

```
#define _TIM2_ETF ((uint8_t) (0x0F << 0))
```

TIM2 External trigger filter [3:0] (in _TIM2_ETR)

Definition at line 1848 of file STM8L10x.h.

5.2.2.529 _TIM2_ETF0

```
#define _TIM2_ETF0 ((uint8_t) (0x01 << 0))
```

TIM2 External trigger filter [0] (in _TIM2_ETR)

Definition at line 1849 of file STM8L10x.h.

5.2.2.530 _TIM2_ETF1

```
#define _TIM2_ETF1 ((uint8_t) (0x01 << 1))
```

TIM2 External trigger filter [1] (in _TIM2_ETR)

Definition at line 1850 of file STM8L10x.h.

5.2.2.531 _TIM2_ETF2

```
#define _TIM2_ETF2 ((uint8_t) (0x01 << 2))
```

TIM2 External trigger filter [2] (in _TIM2_ETR)

Definition at line 1851 of file STM8L10x.h.

5.2.2.532 _TIM2_ETF3

```
#define _TIM2_ETF3 ((uint8_t) (0x01 << 3))
```

TIM2 External trigger filter [3] (in _TIM2_ETR)

Definition at line 1852 of file STM8L10x.h.

5.2.2.533 _TIM2_ETP

```
#define _TIM2_ETP ((uint8_t) (0x01 << 7))
```

TIM2 External trigger polarity [0] (in _TIM2_ETR)

Definition at line 1857 of file STM8L10x.h.

5.2.2.534 _TIM2_ETPS

```
#define _TIM2_ETPS ((uint8_t) (0x03 << 4))
```

TIM2 External trigger prescaler [1:0] (in _TIM2_ETR)

Definition at line 1853 of file STM8L10x.h.

5.2.2.535 _TIM2_ETPS0

```
#define _TIM2_ETPS0 ((uint8_t) (0x01 << 4))
```

TIM2 External trigger prescaler [0] (in _TIM2_ETR)

Definition at line 1854 of file STM8L10x.h.

5.2.2.536 _TIM2_ETPS1

```
#define _TIM2_ETPS1 ((uint8_t) (0x01 << 5))
```

TIM2 External trigger prescaler [1] (in _TIM2_ETR)

Definition at line 1855 of file STM8L10x.h.

5.2.2.537 _TIM2_ETR

```
#define _TIM2_ETR _SFR(uint8_t, TIM2_AddressBase+0x03)
```

TIM2 External trigger register.

Definition at line 1773 of file STM8L10x.h.

5.2.2.538 _TIM2_ETR_RESET_VALUE

```
#define _TIM2_ETR_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 External trigger register reset value.

Definition at line 1797 of file STM8L10x.h.

5.2.2.539 _TIM2_IC1F

```
#define _TIM2_IC1F ((uint8_t) (0x0F << 4))
```

TIM2 Output compare 1 mode [3:0] (in _TIM2_CCMR1)

Definition at line 1906 of file STM8L10x.h.

5.2.2.540 _TIM2_IC1F0

```
#define _TIM2_IC1F0 ((uint8_t) (0x01 << 4))
```

TIM2 Input capture 1 filter [0] (in _TIM2_CCMR1)

Definition at line 1907 of file STM8L10x.h.

5.2.2.541 _TIM2_IC1F1

```
#define _TIM2_IC1F1 ((uint8_t) (0x01 << 5))
```

TIM2 Input capture 1 filter [1] (in _TIM2_CCMR1)

Definition at line 1908 of file STM8L10x.h.

5.2.2.542 _TIM2_IC1F2

```
#define _TIM2_IC1F2 ((uint8_t) (0x01 << 6))
```

TIM2 Input capture 1 filter [2] (in _TIM2_CCMR1)

Definition at line 1909 of file STM8L10x.h.

5.2.2.543 _TIM2_IC1F3

```
#define _TIM2_IC1F3 ((uint8_t) (0x01 << 7))
```

TIM2 Input capture 1 filter [3] (in _TIM2_CCMR1)

Definition at line 1910 of file STM8L10x.h.

5.2.2.544 _TIM2_IC1PSC

```
#define _TIM2_IC1PSC ((uint8_t) (0x03 << 2))
```

TIM2 Input capture 1 prescaler [1:0] (in _TIM2_CCMR1)

Definition at line 1903 of file STM8L10x.h.

5.2.2.545 _TIM2_IC1PSC0

```
#define _TIM2_IC1PSC0 ((uint8_t) (0x01 << 2))
```

TIM2 Input capture 1 prescaler [0] (in _TIM2_CCMR1)

Definition at line 1904 of file STM8L10x.h.

5.2.2.546 _TIM2_IC1PSC1

```
#define _TIM2_IC1PSC1 ((uint8_t) (0x01 << 3))
```

TIM2 Input capture 1 prescaler [1] (in _TIM2_CCMR1)

Definition at line 1905 of file STM8L10x.h.

5.2.2.547 _TIM2_IC2F

```
#define _TIM2_IC2F ((uint8_t) (0x0F << 4))
```

TIM2 Output compare 2 mode [3:0] (in _TIM2_CCMR2)

Definition at line 1929 of file STM8L10x.h.

5.2.2.548 _TIM2_IC2F0

```
#define _TIM2_IC2F0 ((uint8_t) (0x01 << 4))
```

TIM2 Input capture 2 filter [0] (in _TIM2_CCMR2)

Definition at line 1930 of file STM8L10x.h.

5.2.2.549 _TIM2_IC2F1

```
#define _TIM2_IC2F1 ((uint8_t) (0x01 << 5))
```

TIM2 Input capture 2 filter [1] (in _TIM2_CCMR2)

Definition at line 1931 of file STM8L10x.h.

5.2.2.550 _TIM2_IC2F2

```
#define _TIM2_IC2F2 ((uint8_t) (0x01 << 6))
```

TIM2 Input capture 2 filter [2] (in _TIM2_CCMR2)

Definition at line 1932 of file STM8L10x.h.

5.2.2.551 _TIM2_IC2F3

```
#define _TIM2_IC2F3 ((uint8_t) (0x01 << 7))
```

TIM2 Input capture 2 filter [3] (in _TIM2_CCMR2)

Definition at line 1933 of file STM8L10x.h.

5.2.2.552 _TIM2_IC2PSC

```
#define _TIM2_IC2PSC ((uint8_t) (0x03 << 2))
```

TIM2 Input capture 2 prescaler [1:0] (in _TIM2_CCMR2)

Definition at line 1926 of file STM8L10x.h.

5.2.2.553 _TIM2_IC2PSC0

```
#define _TIM2_IC2PSC0 ((uint8_t) (0x01 << 2))
```

TIM2 Input capture 2 prescaler [0] (in _TIM2_CCMR2)

Definition at line 1927 of file STM8L10x.h.

5.2.2.554 _TIM2_IC2PSC1

```
#define _TIM2_IC2PSC1 ((uint8_t) (0x01 << 3))
```

TIM2 Input capture 2 prescaler [1] (in _TIM2_CCMR2)

Definition at line 1928 of file STM8L10x.h.

5.2.2.555 _TIM2_IER

```
#define _TIM2_IER _SFR(uint8_t, TIM2_AddressBase+0x04)
```

TIM2 interrupt enable register.

Definition at line 1774 of file STM8L10x.h.

5.2.2.556 _TIM2_IER_RESET_VALUE

```
#define _TIM2_IER_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 interrupt enable register reset value.

Definition at line 1798 of file STM8L10x.h.

5.2.2.557 _TIM2_LOCK

```
#define _TIM2_LOCK ((int8_t) (0x03 << 0))
```

TIM2 Lock configuration [1:0] (in _TIM2_BKR)

Definition at line 1951 of file STM8L10x.h.

5.2.2.558 _TIM2_LOCK0

```
#define _TIM2_LOCK0 ((uint8_t) (0x01 << 0))
```

TIM2 Lock configuration [0] (in _TIM2_BKR)

Definition at line 1952 of file STM8L10x.h.

5.2.2.559 _TIM2_LOCK1

```
#define _TIM2_LOCK1 ((uint8_t) (0x01 << 1))
```

TIM2 Lock configuration [1] (in _TIM2_BKR)

Definition at line 1953 of file STM8L10x.h.

5.2.2.560 _TIM2_MMS

```
#define _TIM2_MMS ((uint8_t) (0x07 << 4))
```

TIM2 Master mode selection [2:0] (in _TIM2_CR2)

Definition at line 1829 of file STM8L10x.h.

5.2.2.561 _TIM2_MMS0

```
#define _TIM2_MMS0 ((uint8_t) (0x01 << 4))
```

TIM2 Master mode selection [0] (in _TIM2_CR2)

Definition at line 1830 of file STM8L10x.h.

5.2.2.562 _TIM2_MMS1

```
#define _TIM2_MMS1 ((uint8_t) (0x01 << 5))
```

TIM2 Master mode selection [1] (in _TIM2_CR2)

Definition at line 1831 of file STM8L10x.h.

5.2.2.563 _TIM2_MMS2

```
#define _TIM2_MMS2 ((uint8_t) (0x01 << 6))
```

TIM2 Master mode selection [2] (in _TIM2_CR2)

Definition at line 1832 of file STM8L10x.h.

5.2.2.564 _TIM2_MOE

```
#define _TIM2_MOE ((uint8_t) (0x01 << 7))
```

TIM2 Main output enable [0] (in _TIM2_BKR)

Definition at line 1959 of file STM8L10x.h.

5.2.2.565 _TIM2_MSM

```
#define _TIM2_MSM ((uint8_t) (0x01 << 7))
```

TIM2 Master/slave mode [0] (in _TIM2_SMCR)

Definition at line 1845 of file STM8L10x.h.

5.2.2.566 _TIM2_OC1FE

```
#define _TIM2_OC1FE ((uint8_t) (0x01 << 2))
```

TIM2 Output compare 1 fast enable [0] (in _TIM2_CCMR1)

Definition at line 1893 of file STM8L10x.h.

5.2.2.567 _TIM2_OC1M

```
#define _TIM2_OC1M ((uint8_t) (0x07 << 4))
```

TIM2 Output compare 1 mode [2:0] (in _TIM2_CCMR1)

Definition at line 1895 of file STM8L10x.h.

5.2.2.568 _TIM2_OC1M0

```
#define _TIM2_OC1M0 ((uint8_t) (0x01 << 4))
```

TIM2 Output compare 1 mode [0] (in _TIM2_CCMR1)

Definition at line 1896 of file STM8L10x.h.

5.2.2.569 _TIM2_OC1M1

```
#define _TIM2_OC1M1 ((uint8_t) (0x01 << 5))
```

TIM2 Output compare 1 mode [1] (in _TIM2_CCMR1)

Definition at line 1897 of file STM8L10x.h.

5.2.2.570 _TIM2_OC1M2

```
#define _TIM2_OC1M2 ((uint8_t) (0x01 << 6))
```

TIM2 Output compare 1 mode [2] (in _TIM2_CCMR1)

Definition at line 1898 of file STM8L10x.h.

5.2.2.571 _TIM2_OC1PE

```
#define _TIM2_OC1PE ((uint8_t) (0x01 << 3))
```

TIM2 Output compare 1 preload enable [0] (in _TIM2_CCMR1)

Definition at line 1894 of file STM8L10x.h.

5.2.2.572 _TIM2_OC2FE

```
#define _TIM2_OC2FE ((uint8_t) (0x01 << 2))
```

TIM2 Output compare 2 fast enable [0] (in _TIM2_CCMR2)

Definition at line 1916 of file STM8L10x.h.

5.2.2.573 _TIM2_OC2M

```
#define _TIM2_OC2M ((uint8_t) (0x07 << 4))
```

TIM2 Output compare 2 mode [2:0] (in _TIM2_CCMR2)

Definition at line 1918 of file STM8L10x.h.

5.2.2.574 _TIM2_OC2M0

```
#define _TIM2_OC2M0 ((uint8_t) (0x01 << 4))
```

TIM2 Output compare 2 mode [0] (in _TIM2_CCMR2)

Definition at line 1919 of file STM8L10x.h.

5.2.2.575 _TIM2_OC2M1

```
#define _TIM2_OC2M1 ((uint8_t) (0x01 << 5))
```

TIM2 Output compare 2 mode [1] (in _TIM2_CCMR2)

Definition at line 1920 of file STM8L10x.h.

5.2.2.576 _TIM2_OC2M2

```
#define _TIM2_OC2M2 ((uint8_t) (0x01 << 6))
```

TIM2 Output compare 2 mode [2] (in _TIM2_CCMR2)

Definition at line 1921 of file STM8L10x.h.

5.2.2.577 _TIM2_OC2PE

```
#define _TIM2_OC2PE ((uint8_t) (0x01 << 3))
```

TIM2 Output compare 2 preload enable [0] (in _TIM2_CCMR2)

Definition at line 1917 of file STM8L10x.h.

5.2.2.578 _TIM2_OIS1

```
#define _TIM2_OIS1 ((uint8_t) (0x01 << 0))
```

TIM2 Output idle state 1 (OC1 output) [0] (in _TIM2_OISR)

Definition at line 1962 of file STM8L10x.h.

5.2.2.579 _TIM2_OIS2

```
#define _TIM2_OIS2 ((uint8_t) (0x01 << 2))
```

TIM2 Output idle state 2 (OC2 output) [0] (in _TIM2_OISR)

Definition at line 1964 of file STM8L10x.h.

5.2.2.580 _TIM2_OISR

```
#define _TIM2_OISR _SFR(uint8_t, TIM2_AddressBase+0x15)
```

TIM2 Output idle state register.

Definition at line 1791 of file STM8L10x.h.

5.2.2.581 _TIM2_OISR_RESET_VALUE

```
#define _TIM2_OISR_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Output idle state register reset value.

Definition at line 1816 of file STM8L10x.h.

5.2.2.582 _TIM2_OPM

```
#define _TIM2_OPM ((uint8_t) (0x01 << 3))
```

TIM2 One-pulse mode [0] (in _TIM2_CR1)

Definition at line 1822 of file STM8L10x.h.

5.2.2.583 _TIM2_OSSI

```
#define _TIM2_OSSI ((uint8_t) (0x01 << 2))
```

TIM2 Off state selection for idle mode [0] (in _TIM2_BKR)

Definition at line 1954 of file STM8L10x.h.

5.2.2.584 _TIM2_PSC

```
#define _TIM2_PSC ((uint8_t) (0x07 << 0))
```

TIM2 prescaler [2:0] (in _TIM2_PSCR)

Definition at line 1944 of file STM8L10x.h.

5.2.2.585 _TIM2_PSC0

```
#define _TIM2_PSC0 ((uint8_t) (0x01 << 0))
```

TIM2 prescaler [0] (in _TIM2_PSCR)

Definition at line 1945 of file STM8L10x.h.

5.2.2.586 _TIM2_PSC1

```
#define _TIM2_PSC1 ((uint8_t) (0x01 << 1))
```

TIM2 prescaler [1] (in _TIM2_PSCR)

Definition at line 1946 of file STM8L10x.h.

5.2.2.587 _TIM2_PSC2

```
#define _TIM2_PSC2 ((uint8_t) (0x01 << 2))
```

TIM2 prescaler [2] (in _TIM2_PSCR)

Definition at line 1947 of file STM8L10x.h.

5.2.2.588 _TIM2_PSCR

```
#define _TIM2_PSCR _SFR(uint8_t, TIM2_AddressBase+0x0D)
```

TIM2 clock prescaler register.

Definition at line 1783 of file STM8L10x.h.

5.2.2.589 _TIM2_PSCR_RESET_VALUE

```
#define _TIM2_PSCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 clock prescaler register reset value.

Definition at line 1807 of file STM8L10x.h.

5.2.2.590 _TIM2_RCR_RESET_VALUE

```
#define _TIM2_RCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Repetition counter reset value.

Definition at line 1810 of file STM8L10x.h.

5.2.2.591 _TIM2_SMCR

```
#define _TIM2_SMCR _SFR(uint8_t, TIM2_AddressBase+0x02)
```

TIM2 Slave mode control register.

Definition at line 1772 of file STM8L10x.h.

5.2.2.592 _TIM2_SMCR_RESET_VALUE

```
#define _TIM2_SMCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 Slave mode control register reset value.

Definition at line 1796 of file STM8L10x.h.

5.2.2.593 _TIM2_SMS

```
#define _TIM2_SMS ((uint8_t) (0x07 << 0))
```

TIM2 Clock/trigger/slave mode selection [2:0] (in _TIM2_SMCR)

Definition at line 1836 of file STM8L10x.h.

5.2.2.594 _TIM2_SMS0

```
#define _TIM2_SMS0 ((uint8_t) (0x01 << 0))
```

TIM2 Clock/trigger/slave mode selection [0] (in _TIM2_SMCR)

Definition at line 1837 of file STM8L10x.h.

5.2.2.595 _TIM2_SMS1

```
#define _TIM2_SMS1 ((uint8_t) (0x01 << 1))
```

TIM2 Clock/trigger/slave mode selection [1] (in _TIM2_SMCR)

Definition at line 1838 of file STM8L10x.h.

5.2.2.596 _TIM2_SMS2

```
#define _TIM2_SMS2 ((uint8_t) (0x01 << 2))
```

TIM2 Clock/trigger/slave mode selection [2] (in _TIM2_SMCR)

Definition at line 1839 of file STM8L10x.h.

5.2.2.597 _TIM2_SR1

```
#define _TIM2_SR1 _SFR(uint8_t, TIM2_AddressBase+0x05)
```

TIM2 status register 1.

Definition at line 1775 of file STM8L10x.h.

5.2.2.598 _TIM2_SR1_RESET_VALUE

```
#define _TIM2_SR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 status register 1 reset value.

Definition at line 1799 of file STM8L10x.h.

5.2.2.599 _TIM2_SR2

```
#define _TIM2_SR2 _SFR(uint8_t, TIM2_AddressBase+0x06)
```

TIM2 status register 2.

Definition at line 1776 of file STM8L10x.h.

5.2.2.600 _TIM2_SR2_RESET_VALUE

```
#define _TIM2_SR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM2 status register 2 reset value.

Definition at line 1800 of file STM8L10x.h.

5.2.2.601 _TIM2_TG

```
#define _TIM2_TG ((uint8_t) (0x01 << 6))
```

TIM2 Trigger generation [0] (in _TIM2_EGR)

Definition at line 1886 of file STM8L10x.h.

5.2.2.602 _TIM2_TIE

```
#define _TIM2_TIE ((uint8_t) (0x01 << 6))
```

TIM2 Trigger interrupt enable [0] (in _TIM2_IER)

Definition at line 1864 of file STM8L10x.h.

5.2.2.603 _TIM2_TIF

```
#define _TIM2_TIF ((uint8_t) (0x01 << 6))
```

TIM2 Trigger interrupt flag [0] (in _TIM2_SR1)

Definition at line 1872 of file STM8L10x.h.

5.2.2.604 _TIM2_TS

```
#define _TIM2_TS ((uint8_t) (0x07 << 4))
```

TIM2 Trigger selection [2:0] (in _TIM2_SMCR)

Definition at line 1841 of file STM8L10x.h.

5.2.2.605 _TIM2_TS0

```
#define _TIM2_TS0 ((uint8_t) (0x01 << 4))
```

TIM2 Trigger selection [0] (in _TIM2_SMCR)

Definition at line 1842 of file STM8L10x.h.

5.2.2.606 _TIM2_TS1

```
#define _TIM2_TS1 ((uint8_t) (0x01 << 5))
```

TIM2 Trigger selection [1] (in _TIM2_SMCR)

Definition at line 1843 of file STM8L10x.h.

5.2.2.607 _TIM2_TS2

```
#define _TIM2_TS2 ((uint8_t) (0x01 << 6))
```

TIM2 Trigger selection [2] (in _TIM2_SMCR)

Definition at line 1844 of file STM8L10x.h.

5.2.2.608 _TIM2_UDIS

```
#define _TIM2_UDIS ((uint8_t) (0x01 << 1))
```

TIM2 Update disable [0] (in _TIM2_CR1)

Definition at line 1820 of file STM8L10x.h.

5.2.2.609 _TIM2_UG

```
#define _TIM2_UG ((uint8_t) (0x01 << 0))
```

TIM2 Update generation [0] (in _TIM2_EGR)

Definition at line 1882 of file STM8L10x.h.

5.2.2.610 _TIM2_UIE

```
#define _TIM2_UIE ((uint8_t) (0x01 << 0))
```

TIM2 Update interrupt enable [0] (in _TIM2_IER)

Definition at line 1860 of file STM8L10x.h.

5.2.2.611 _TIM2_UIF

```
#define _TIM2_UIF ((uint8_t) (0x01 << 0))
```

TIM2 Update interrupt flag [0] (in _TIM2_SR1)

Definition at line 1868 of file STM8L10x.h.

5.2.2.612 _TIM2_URS

```
#define _TIM2_URS ((uint8_t) (0x01 << 2))
```

TIM2 Update request source [0] (in _TIM2_CR1)

Definition at line 1821 of file STM8L10x.h.

5.2.2.613 _TIM3

```
#define _TIM3 _SFR(TIM3_3_t, TIM3_AddressBase)
```

TIM3 struct/bit access.

Definition at line 1972 of file STM8L10x.h.

5.2.2.614 _TIM3_AOE

```
#define _TIM3_AOE ((uint8_t) (0x01 << 6))
```

TIM3 Automatic output enable [0] (in _TIM3_BKR)

Definition at line 2161 of file STM8L10x.h.

5.2.2.615 _TIM3_ARPE

```
#define _TIM3_ARPE ((uint8_t) (0x01 << 7))
```

TIM3 Auto-reload preload enable [0] (in _TIM3_CR1)

Definition at line 2028 of file STM8L10x.h.

5.2.2.616 _TIM3_ARRH

```
#define _TIM3_ARRH _SFR(uint8_t, TIM3_AddressBase+0x0E)
```

TIM3 auto-reload register high byte.

Definition at line 1987 of file STM8L10x.h.

5.2.2.617 _TIM3_ARRH_RESET_VALUE

```
#define _TIM3_ARRH_RESET_VALUE ((uint8_t) 0xFF)
```

TIM3 auto-reload register high byte reset value.

Definition at line 2011 of file STM8L10x.h.

5.2.2.618 _TIM3_ARRL

```
#define _TIM3_ARRL _SFR(uint8_t, TIM3_AddressBase+0x0F)
```

TIM3 auto-reload register low byte.

Definition at line 1988 of file STM8L10x.h.

5.2.2.619 _TIM3_ARRL_RESET_VALUE

```
#define _TIM3_ARRL_RESET_VALUE ((uint8_t) 0xFF)
```

TIM3 auto-reload register low byte reset value.

Definition at line 2012 of file STM8L10x.h.

5.2.2.620 _TIM3_BG

```
#define _TIM3_BG ((uint8_t) (0x01 << 7))
```

TIM3 Break generation [0] (in _TIM3_EGR)

Definition at line 2090 of file STM8L10x.h.

5.2.2.621 _TIM3_BIE

```
#define _TIM3_BIE ((uint8_t) (0x01 << 7))
```

TIM3 Break interrupt enable [0] (in _TIM3_IER)

Definition at line 2068 of file STM8L10x.h.

5.2.2.622 _TIM3_BIF

```
#define _TIM3_BIF ((uint8_t) (0x01 << 7))
```

TIM3 Break interrupt flag [0] (in _TIM3_SR1)

Definition at line 2076 of file STM8L10x.h.

5.2.2.623 _TIM3_BKE

```
#define _TIM3_BKE ((uint8_t) (0x01 << 4))
```

TIM3 Break enable [0] (in _TIM3_BKR)

Definition at line 2159 of file STM8L10x.h.

5.2.2.624 _TIM3_BKP

```
#define _TIM3_BKP ((uint8_t) (0x01 << 5))
```

TIM3 Break polarity [0] (in _TIM3_BKR)

Definition at line 2160 of file STM8L10x.h.

5.2.2.625 _TIM3_BKR

```
#define _TIM3_BKR _SFR(uint8_t, TIM3_AddressBase+0x14)
```

TIM3 Break register.

Definition at line 1993 of file STM8L10x.h.

5.2.2.626 _TIM3_BKR_RESET_VALUE

```
#define _TIM3_BKR_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 Break register reset value.

Definition at line 2018 of file STM8L10x.h.

5.2.2.627 _TIM3_CC1E

```
#define _TIM3_CC1E ((uint8_t) (0x01 << 0))
```

TIM3 Capture/compare 1 output enable [0] (in _TIM3_CCER1)

Definition at line 2139 of file STM8L10x.h.

5.2.2.628 _TIM3_CC1G

```
#define _TIM3_CC1G ((uint8_t) (0x01 << 1))
```

TIM3 Capture/compare 1 generation [0] (in _TIM3_EGR)

Definition at line 2086 of file STM8L10x.h.

5.2.2.629 _TIM3_CC1IE

```
#define _TIM3_CC1IE ((uint8_t) (0x01 << 1))
```

TIM3 Capture/compare 1 interrupt enable [0] (in _TIM3_IER)

Definition at line 2064 of file STM8L10x.h.

5.2.2.630 _TIM3_CC1IF

```
#define _TIM3_CC1IF ((uint8_t) (0x01 << 1))
```

TIM3 Capture/compare 1 interrupt flag [0] (in _TIM3_SR1)

Definition at line 2072 of file STM8L10x.h.

5.2.2.631 _TIM3_CC1OF

```
#define _TIM3_CC1OF ((uint8_t) (0x01 << 1))
```

TIM3 Capture/compare 1 overcapture flag [0] (in _TIM3_SR2)

Definition at line 2080 of file STM8L10x.h.

5.2.2.632 _TIM3_CC1P

```
#define _TIM3_CC1P ((uint8_t) (0x01 << 1))
```

TIM3 Capture/compare 1 output polarity [0] (in _TIM3_CCER1)

Definition at line 2140 of file STM8L10x.h.

5.2.2.633 _TIM3_CC1S

```
#define _TIM3_CC1S ((uint8_t) (0x03 << 0))
```

TIM3 Compare 1 selection [1:0] (in _TIM3_CCMR1)

Definition at line 2093 of file STM8L10x.h.

5.2.2.634 _TIM3_CC1S0

```
#define _TIM3_CC1S0 ((uint8_t) (0x01 << 0))
```

TIM3 Compare 1 selection [0] (in _TIM3_CCMR1)

Definition at line 2094 of file STM8L10x.h.

5.2.2.635 _TIM3_CC1S1

```
#define _TIM3_CC1S1 ((uint8_t) (0x01 << 1))
```

TIM3 Compare 1 selection [1] (in _TIM3_CCMR1)

Definition at line 2095 of file STM8L10x.h.

5.2.2.636 _TIM3_CC2E

```
#define _TIM3_CC2E ((uint8_t) (0x01 << 4))
```

TIM3 Capture/compare 2 output enable [0] (in _TIM3_CCER1)

Definition at line 2142 of file STM8L10x.h.

5.2.2.637 _TIM3_CC2G

```
#define _TIM3_CC2G ((uint8_t) (0x01 << 2))
```

TIM3 Capture/compare 2 generation [0] (in _TIM3_EGR)

Definition at line 2087 of file STM8L10x.h.

5.2.2.638 _TIM3_CC2IE

```
#define _TIM3_CC2IE ((uint8_t) (0x01 << 2))
```

TIM3 Capture/compare 2 interrupt enable [0] (in _TIM3_IER)

Definition at line 2065 of file STM8L10x.h.

5.2.2.639 _TIM3_CC2IF

```
#define _TIM3_CC2IF ((uint8_t) (0x01 << 2))
```

TIM3 Capture/compare 2 interrupt flag [0] (in _TIM3_SR1)

Definition at line 2073 of file STM8L10x.h.

5.2.2.640 _TIM3_CC2OF

```
#define _TIM3_CC2OF ((uint8_t) (0x01 << 2))
```

TIM3 Capture/compare 2 overcapture flag [0] (in _TIM3_SR2)

Definition at line 2081 of file STM8L10x.h.

5.2.2.641 _TIM3_CC2P

```
#define _TIM3_CC2P ((uint8_t) (0x01 << 5))
```

TIM3 Capture/compare 2 output polarity [0] (in _TIM3_CCER1)

Definition at line 2143 of file STM8L10x.h.

5.2.2.642 _TIM3_CC2S

```
#define _TIM3_CC2S ((uint8_t) (0x03 << 0))
```

TIM3 Compare 2 selection [1:0] (in _TIM3_CCMR2)

Definition at line 2116 of file STM8L10x.h.

5.2.2.643 _TIM3_CC2S0

```
#define _TIM3_CC2S0 ((uint8_t) (0x01 << 0))
```

TIM3 Compare 2 selection [0] (in _TIM3_CCMR2)

Definition at line 2117 of file STM8L10x.h.

5.2.2.644 _TIM3_CC2S1

```
#define _TIM3_CC2S1 ((uint8_t) (0x01 << 1))
```

TIM3 Compare 2 selection [1] (in _TIM3_CCMR2)

Definition at line 2118 of file STM8L10x.h.

5.2.2.645 _TIM3_CCER1

```
#define _TIM3_CCER1 _SFR(uint8_t, TIM3_AddressBase+0x0A)
```

TIM3 Capture/compare enable register 1.

Definition at line 1983 of file STM8L10x.h.

5.2.2.646 _TIM3_CCER1_RESET_VALUE

```
#define _TIM3_CCER1_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 Capture/compare enable register 1 reset value.

Definition at line 2007 of file STM8L10x.h.

5.2.2.647 _TIM3_CCMR1

```
#define _TIM3_CCMR1 _SFR(uint8_t, TIM3_AddressBase+0x08)
```

TIM3 Capture/compare mode register 1.

Definition at line 1981 of file STM8L10x.h.

5.2.2.648 _TIM3_CCMR1_RESET_VALUE

```
#define _TIM3_CCMR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 Capture/compare mode register 1 reset value.

Definition at line 2005 of file STM8L10x.h.

5.2.2.649 _TIM3_CCMR2

```
#define _TIM3_CCMR2 _SFR(uint8_t, TIM3_AddressBase+0x09)
```

TIM3 Capture/compare mode register 2.

Definition at line 1982 of file STM8L10x.h.

5.2.2.650 _TIM3_CCMR2_RESET_VALUE

```
#define _TIM3_CCMR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 Capture/compare mode register 2 reset value.

Definition at line 2006 of file STM8L10x.h.

5.2.2.651 _TIM3_CCR1H

```
#define _TIM3_CCR1H _SFR(uint8_t, TIM3_AddressBase+0x10)
```

TIM3 16-bit capture/compare value 1 high byte.

Definition at line 1989 of file STM8L10x.h.

5.2.2.652 _TIM3_CCR1H_RESET_VALUE

```
#define _TIM3_CCR1H_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 16-bit capture/compare value 1 high byte reset value.

Definition at line 2014 of file STM8L10x.h.

5.2.2.653 _TIM3_CCR1L

```
#define _TIM3_CCR1L _SFR(uint8_t, TIM3_AddressBase+0x11)
```

TIM3 16-bit capture/compare value 1 low byte.

Definition at line 1990 of file STM8L10x.h.

5.2.2.654 _TIM3_CCR1L_RESET_VALUE

```
#define _TIM3_CCR1L_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 16-bit capture/compare value 1 low byte reset value.

Definition at line 2015 of file STM8L10x.h.

5.2.2.655 _TIM3_CCR2H

```
#define _TIM3_CCR2H _SFR(uint8_t, TIM3_AddressBase+0x12)
```

TIM3 16-bit capture/compare value 2 high byte.

Definition at line 1991 of file STM8L10x.h.

5.2.2.656 _TIM3_CCR2H_RESET_VALUE

```
#define _TIM3_CCR2H_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 16-bit capture/compare value 2 high byte reset value.

Definition at line 2016 of file STM8L10x.h.

5.2.2.657 _TIM3_CCR2L

```
#define _TIM3_CCR2L _SFR(uint8_t, TIM3_AddressBase+0x13)
```

TIM3 16-bit capture/compare value 2 low byte.

Definition at line 1992 of file STM8L10x.h.

5.2.2.658 _TIM3_CCR2L_RESET_VALUE

```
#define _TIM3_CCR2L_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 16-bit capture/compare value 2 low byte reset value.

Definition at line 2017 of file STM8L10x.h.

5.2.2.659 _TIM3_CEN

```
#define _TIM3_CEN ((uint8_t) (0x01 << 0))
```

TIM3 Counter enable [0] (in _TIM3_CR1)

Definition at line 2022 of file STM8L10x.h.

5.2.2.660 _TIM3_CNTRH

```
#define _TIM3_CNTRH _SFR(uint8_t, TIM3_AddressBase+0x0B)
```

TIM3 counter register high byte.

Definition at line 1984 of file STM8L10x.h.

5.2.2.661 _TIM3_CNTRH_RESET_VALUE

```
#define _TIM3_CNTRH_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 counter register high byte reset value.

Definition at line 2008 of file STM8L10x.h.

5.2.2.662 _TIM3_CNTRL

```
#define _TIM3_CNTRL _SFR(uint8_t, TIM3_AddressBase+0x0C)
```

TIM3 counter register low byte.

Definition at line 1985 of file STM8L10x.h.

5.2.2.663 _TIM3_CNTRL_RESET_VALUE

```
#define _TIM3_CNTRL_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 counter register low byte reset value.

Definition at line 2009 of file STM8L10x.h.

5.2.2.664 _TIM3_CR1

```
#define _TIM3_CR1 _SFR(uint8_t, TIM3_AddressBase+0x00)
```

TIM3 control register 1.

Definition at line 1973 of file STM8L10x.h.

5.2.2.665 _TIM3_CR1_RESET_VALUE

```
#define _TIM3_CR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 control register 1 reset value.

Definition at line 1997 of file STM8L10x.h.

5.2.2.666 _TIM3_CR2

```
#define _TIM3_CR2 _SFR(uint8_t, TIM3_AddressBase+0x01)
```

TIM3 control register 2.

Definition at line 1974 of file STM8L10x.h.

5.2.2.667 _TIM3_CR2_RESET_VALUE

```
#define _TIM3_CR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 control register 2 reset value.

Definition at line 1998 of file STM8L10x.h.

5.2.2.668 _TIM3_DIR

```
#define _TIM3_DIR ((uint8_t) (0x01 << 4))
```

TIM3 Direction [0] (in _TIM3_CR1)

Definition at line 2026 of file STM8L10x.h.

5.2.2.669 _TIM3_ECE

```
#define _TIM3_ECE ((uint8_t) (0x01 << 6))
```

TIM3 External clock enable [0] (in _TIM3_ETR)

Definition at line 2059 of file STM8L10x.h.

5.2.2.670 _TIM3_EGR

```
#define _TIM3_EGR _SFR(uint8_t, TIM3_AddressBase+0x07)
```

TIM3 Event generation register.

Definition at line 1980 of file STM8L10x.h.

5.2.2.671 _TIM3_EGR_RESET_VALUE

```
#define _TIM3_EGR_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 Event generation register reset value.

Definition at line 2004 of file STM8L10x.h.

5.2.2.672 _TIM3_ETF

```
#define _TIM3_ETF ((uint8_t) (0x0F << 0))
```

TIM3 External trigger filter [3:0] (in _TIM3_ETR)

Definition at line 2051 of file STM8L10x.h.

5.2.2.673 _TIM3_ETF0

```
#define _TIM3_ETF0 ((uint8_t) (0x01 << 0))
```

TIM3 External trigger filter [0] (in _TIM3_ETR)

Definition at line 2052 of file STM8L10x.h.

5.2.2.674 _TIM3_ETF1

```
#define _TIM3_ETF1 ((uint8_t) (0x01 << 1))
```

TIM3 External trigger filter [1] (in _TIM3_ETR)

Definition at line 2053 of file STM8L10x.h.

5.2.2.675 _TIM3_ETF2

```
#define _TIM3_ETF2 ((uint8_t) (0x01 << 2))
```

TIM3 External trigger filter [2] (in _TIM3_ETR)

Definition at line 2054 of file STM8L10x.h.

5.2.2.676 _TIM3_ETF3

```
#define _TIM3_ETF3 ((uint8_t) (0x01 << 3))
```

TIM3 External trigger filter [3] (in _TIM3_ETR)

Definition at line 2055 of file STM8L10x.h.

5.2.2.677 _TIM3_ETP

```
#define _TIM3_ETP ((uint8_t) (0x01 << 7))
```

TIM3 External trigger polarity [0] (in _TIM3_ETR)

Definition at line 2060 of file STM8L10x.h.

5.2.2.678 _TIM3_ETPS

```
#define _TIM3_ETPS ((uint8_t) (0x03 << 4))
```

TIM3 External trigger prescaler [1:0] (in _TIM3_ETR)

Definition at line 2056 of file STM8L10x.h.

5.2.2.679 _TIM3_ETPS0

```
#define _TIM3_ETPS0 ((uint8_t) (0x01 << 4))
```

TIM3 External trigger prescaler [0] (in _TIM3_ETR)

Definition at line 2057 of file STM8L10x.h.

5.2.2.680 _TIM3_ETPS1

```
#define _TIM3_ETPS1 ((uint8_t) (0x01 << 5))
```

TIM3 External trigger prescaler [1] (in _TIM3_ETR)

Definition at line 2058 of file STM8L10x.h.

5.2.2.681 _TIM3_ETR

```
#define _TIM3_ETR _SFR(uint8_t, TIM3_AddressBase+0x03)
```

TIM3 External trigger register.

Definition at line 1976 of file STM8L10x.h.

5.2.2.682 _TIM3_ETR_RESET_VALUE

```
#define _TIM3_ETR_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 External trigger register reset value.

Definition at line 2000 of file STM8L10x.h.

5.2.2.683 _TIM3_IC1F

```
#define _TIM3_IC1F ((uint8_t) (0x0F << 4))
```

TIM3 Output compare 1 mode [3:0] (in _TIM3_CCMR1)

Definition at line 2109 of file STM8L10x.h.

5.2.2.684 _TIM3_IC1F0

```
#define _TIM3_IC1F0 ((uint8_t) (0x01 << 4))
```

TIM3 Input capture 1 filter [0] (in _TIM3_CCMR1)

Definition at line 2110 of file STM8L10x.h.

5.2.2.685 _TIM3_IC1F1

```
#define _TIM3_IC1F1 ((uint8_t) (0x01 << 5))
```

TIM3 Input capture 1 filter [1] (in _TIM3_CCMR1)

Definition at line 2111 of file STM8L10x.h.

5.2.2.686 _TIM3_IC1F2

```
#define _TIM3_IC1F2 ((uint8_t) (0x01 << 6))
```

TIM3 Input capture 1 filter [2] (in _TIM3_CCMR1)

Definition at line 2112 of file STM8L10x.h.

5.2.2.687 _TIM3_IC1F3

```
#define _TIM3_IC1F3 ((uint8_t) (0x01 << 7))
```

TIM3 Input capture 1 filter [3] (in _TIM3_CCMR1)

Definition at line 2113 of file STM8L10x.h.

5.2.2.688 _TIM3_IC1PSC

```
#define _TIM3_IC1PSC ((uint8_t) (0x03 << 2))
```

TIM3 Input capture 1 prescaler [1:0] (in _TIM3_CCMR1)

Definition at line 2106 of file STM8L10x.h.

5.2.2.689 _TIM3_IC1PSC0

```
#define _TIM3_IC1PSC0 ((uint8_t) (0x01 << 2))
```

TIM3 Input capture 1 prescaler [0] (in _TIM3_CCMR1)

Definition at line 2107 of file STM8L10x.h.

5.2.2.690 _TIM3_IC1PSC1

```
#define _TIM3_IC1PSC1 ((uint8_t) (0x01 << 3))
```

TIM3 Input capture 1 prescaler [1] (in _TIM3_CCMR1)

Definition at line 2108 of file STM8L10x.h.

5.2.2.691 _TIM3_IC2F

```
#define _TIM3_IC2F ((uint8_t) (0x0F << 4))
```

TIM3 Output compare 2 mode [3:0] (in _TIM3_CCMR2)

Definition at line 2132 of file STM8L10x.h.

5.2.2.692 _TIM3_IC2F0

```
#define _TIM3_IC2F0 ((uint8_t) (0x01 << 4))
```

TIM3 Input capture 2 filter [0] (in _TIM3_CCMR2)

Definition at line 2133 of file STM8L10x.h.

5.2.2.693 _TIM3_IC2F1

```
#define _TIM3_IC2F1 ((uint8_t) (0x01 << 5))
```

TIM3 Input capture 2 filter [1] (in _TIM3_CCMR2)

Definition at line 2134 of file STM8L10x.h.

5.2.2.694 _TIM3_IC2F2

```
#define _TIM3_IC2F2 ((uint8_t) (0x01 << 6))
```

TIM3 Input capture 2 filter [2] (in _TIM3_CCMR2)

Definition at line 2135 of file STM8L10x.h.

5.2.2.695 _TIM3_IC2F3

```
#define _TIM3_IC2F3 ((uint8_t) (0x01 << 7))
```

TIM3 Input capture 2 filter [3] (in _TIM3_CCMR2)

Definition at line 2136 of file STM8L10x.h.

5.2.2.696 _TIM3_IC2PSC

```
#define _TIM3_IC2PSC ((uint8_t) (0x03 << 2))
```

TIM3 Input capture 2 prescaler [1:0] (in _TIM3_CCMR2)

Definition at line 2129 of file STM8L10x.h.

5.2.2.697 _TIM3_IC2PSC0

```
#define _TIM3_IC2PSC0 ((uint8_t) (0x01 << 2))
```

TIM3 Input capture 2 prescaler [0] (in _TIM3_CCMR2)

Definition at line 2130 of file STM8L10x.h.

5.2.2.698 _TIM3_IC2PSC1

```
#define _TIM3_IC2PSC1 ((uint8_t) (0x01 << 3))
```

TIM3 Input capture 2 prescaler [1] (in _TIM3_CCMR2)

Definition at line 2131 of file STM8L10x.h.

5.2.2.699 _TIM3_IER

```
#define _TIM3_IER _SFR(uint8_t, TIM3_AddressBase+0x04)
```

TIM3 interrupt enable register.

Definition at line 1977 of file STM8L10x.h.

5.2.2.700 _TIM3_IER_RESET_VALUE

```
#define _TIM3_IER_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 interrupt enable register reset value.

Definition at line 2001 of file STM8L10x.h.

5.2.2.701 _TIM3_LOCK

```
#define _TIM3_LOCK ((int8_t) (0x03 << 0))
```

TIM3 Lock configuration [1:0] (in _TIM3_BKR)

Definition at line 2154 of file STM8L10x.h.

5.2.2.702 _TIM3_LOCK0

```
#define _TIM3_LOCK0 ((uint8_t) (0x01 << 0))
```

TIM3 Lock configuration [0] (in _TIM3_BKR)

Definition at line 2155 of file STM8L10x.h.

5.2.2.703 _TIM3_LOCK1

```
#define _TIM3_LOCK1 ((uint8_t) (0x01 << 1))
```

TIM3 Lock configuration [1] (in _TIM3_BKR)

Definition at line 2156 of file STM8L10x.h.

5.2.2.704 _TIM3_MMS

```
#define _TIM3_MMS ((uint8_t) (0x07 << 4))
```

TIM3 Master mode selection [2:0] (in _TIM3_CR2)

Definition at line 2032 of file STM8L10x.h.

5.2.2.705 _TIM3_MMS0

```
#define _TIM3_MMS0 ((uint8_t) (0x01 << 4))
```

TIM3 Master mode selection [0] (in _TIM3_CR2)

Definition at line 2033 of file STM8L10x.h.

5.2.2.706 _TIM3_MMS1

```
#define _TIM3_MMS1 ((uint8_t) (0x01 << 5))
```

TIM3 Master mode selection [1] (in _TIM3_CR2)

Definition at line 2034 of file STM8L10x.h.

5.2.2.707 _TIM3_MMS2

```
#define _TIM3_MMS2 ((uint8_t) (0x01 << 6))
```

TIM3 Master mode selection [2] (in _TIM3_CR2)

Definition at line 2035 of file STM8L10x.h.

5.2.2.708 _TIM3_MOE

```
#define _TIM3_MOE ((uint8_t) (0x01 << 7))
```

TIM3 Main output enable [0] (in _TIM3_BKR)

Definition at line 2162 of file STM8L10x.h.

5.2.2.709 _TIM3_MSM

```
#define _TIM3_MSM ((uint8_t) (0x01 << 7))
```

TIM3 Master/slave mode [0] (in _TIM3_SMCR)

Definition at line 2048 of file STM8L10x.h.

5.2.2.710 _TIM3_OC1FE

```
#define _TIM3_OC1FE ((uint8_t) (0x01 << 2))
```

TIM3 Output compare 1 fast enable [0] (in _TIM3_CCMR1)

Definition at line 2096 of file STM8L10x.h.

5.2.2.711 _TIM3_OC1M

```
#define _TIM3_OC1M ((uint8_t) (0x07 << 4))
```

TIM3 Output compare 1 mode [2:0] (in _TIM3_CCMR1)

Definition at line 2098 of file STM8L10x.h.

5.2.2.712 _TIM3_OC1M0

```
#define _TIM3_OC1M0 ((uint8_t) (0x01 << 4))
```

TIM3 Output compare 1 mode [0] (in _TIM3_CCMR1)

Definition at line 2099 of file STM8L10x.h.

5.2.2.713 _TIM3_OC1M1

```
#define _TIM3_OC1M1 ((uint8_t) (0x01 << 5))
```

TIM3 Output compare 1 mode [1] (in _TIM3_CCMR1)

Definition at line 2100 of file STM8L10x.h.

5.2.2.714 _TIM3_OC1M2

```
#define _TIM3_OC1M2 ((uint8_t) (0x01 << 6))
```

TIM3 Output compare 1 mode [2] (in _TIM3_CCMR1)

Definition at line 2101 of file STM8L10x.h.

5.2.2.715 _TIM3_OC1PE

```
#define _TIM3_OC1PE ((uint8_t) (0x01 << 3))
```

TIM3 Output compare 1 preload enable [0] (in _TIM3_CCMR1)

Definition at line 2097 of file STM8L10x.h.

5.2.2.716 _TIM3_OC2FE

```
#define _TIM3_OC2FE ((uint8_t) (0x01 << 2))
```

TIM3 Output compare 2 fast enable [0] (in _TIM3_CCMR2)

Definition at line 2119 of file STM8L10x.h.

5.2.2.717 _TIM3_OC2M

```
#define _TIM3_OC2M ((uint8_t) (0x07 << 4))
```

TIM3 Output compare 2 mode [2:0] (in _TIM3_CCMR2)

Definition at line 2121 of file STM8L10x.h.

5.2.2.718 _TIM3_OC2M0

```
#define _TIM3_OC2M0 ((uint8_t) (0x01 << 4))
```

TIM3 Output compare 2 mode [0] (in _TIM3_CCMR2)

Definition at line 2122 of file STM8L10x.h.

5.2.2.719 _TIM3_OC2M1

```
#define _TIM3_OC2M1 ((uint8_t) (0x01 << 5))
```

TIM3 Output compare 2 mode [1] (in _TIM3_CCMR2)

Definition at line 2123 of file STM8L10x.h.

5.2.2.720 _TIM3_OC2M2

```
#define _TIM3_OC2M2 ((uint8_t) (0x01 << 6))
```

TIM3 Output compare 2 mode [2] (in _TIM3_CCMR2)

Definition at line 2124 of file STM8L10x.h.

5.2.2.721 _TIM3_OC2PE

```
#define _TIM3_OC2PE ((uint8_t) (0x01 << 3))
```

TIM3 Output compare 2 preload enable [0] (in _TIM3_CCMR2)

Definition at line 2120 of file STM8L10x.h.

5.2.2.722 _TIM3_OIS1

```
#define _TIM3_OIS1 ((uint8_t) (0x01 << 0))
```

TIM3 Output idle state 1 (OC1 output) [0] (in _TIM3_OISR)

Definition at line 2165 of file STM8L10x.h.

5.2.2.723 _TIM3_OIS2

```
#define _TIM3_OIS2 ((uint8_t) (0x01 << 2))
```

TIM3 Output idle state 2 (OC2 output) [0] (in _TIM3_OISR)

Definition at line 2167 of file STM8L10x.h.

5.2.2.724 _TIM3_OISR

```
#define _TIM3_OISR _SFR(uint8_t, TIM3_AddressBase+0x15)
```

TIM3 Output idle state register.

Definition at line 1994 of file STM8L10x.h.

5.2.2.725 _TIM3_OISR_RESET_VALUE

```
#define _TIM3_OISR_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 Output idle state register reset value.

Definition at line 2019 of file STM8L10x.h.

5.2.2.726 _TIM3_OPM

```
#define _TIM3_OPM ((uint8_t) (0x01 << 3))
```

TIM3 One-pulse mode [0] (in _TIM3_CR1)

Definition at line 2025 of file STM8L10x.h.

5.2.2.727 _TIM3_OSSI

```
#define _TIM3_OSSI ((uint8_t) (0x01 << 2))
```

TIM3 Off state selection for idle mode [0] (in _TIM3_BKR)

Definition at line 2157 of file STM8L10x.h.

5.2.2.728 _TIM3_PSC

```
#define _TIM3_PSC ((uint8_t) (0x07 << 0))
```

TIM3 prescaler [2:0] (in _TIM3_PSCR)

Definition at line 2147 of file STM8L10x.h.

5.2.2.729 _TIM3_PSC0

```
#define _TIM3_PSC0 ((uint8_t) (0x01 << 0))
```

TIM3 prescaler [0] (in _TIM3_PSCR)

Definition at line 2148 of file STM8L10x.h.

5.2.2.730 _TIM3_PSC1

```
#define _TIM3_PSC1 ((uint8_t) (0x01 << 1))
```

TIM3 prescaler [1] (in _TIM3_PSCR)

Definition at line 2149 of file STM8L10x.h.

5.2.2.731 _TIM3_PSC2

```
#define _TIM3_PSC2 ((uint8_t) (0x01 << 2))
```

TIM3 prescaler [2] (in _TIM3_PSCR)

Definition at line 2150 of file STM8L10x.h.

5.2.2.732 _TIM3_PSCR

```
#define _TIM3_PSCR _SFR(uint8_t, TIM3_AddressBase+0x0D)
```

TIM3 clock prescaler register.

Definition at line 1986 of file STM8L10x.h.

5.2.2.733 _TIM3_PSCR_RESET_VALUE

```
#define _TIM3_PSCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 clock prescaler register reset value.

Definition at line 2010 of file STM8L10x.h.

5.2.2.734 _TIM3_RCR_RESET_VALUE

```
#define _TIM3_RCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 Repetition counter reset value.

Definition at line 2013 of file STM8L10x.h.

5.2.2.735 _TIM3_SMCR

```
#define _TIM3_SMCR _SFR(uint8_t, TIM3_AddressBase+0x02)
```

TIM3 Slave mode control register.

Definition at line 1975 of file STM8L10x.h.

5.2.2.736 _TIM3_SMCR_RESET_VALUE

```
#define _TIM3_SMCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 Slave mode control register reset value.

Definition at line 1999 of file STM8L10x.h.

5.2.2.737 _TIM3_SMS

```
#define _TIM3_SMS ((uint8_t) (0x07 << 0))
```

TIM3 Clock/trigger/slave mode selection [2:0] (in _TIM3_SMCR)

Definition at line 2039 of file STM8L10x.h.

5.2.2.738 _TIM3_SMS0

```
#define _TIM3_SMS0 ((uint8_t) (0x01 << 0))
```

TIM3 Clock/trigger/slave mode selection [0] (in _TIM3_SMCR)

Definition at line 2040 of file STM8L10x.h.

5.2.2.739 _TIM3_SMS1

```
#define _TIM3_SMS1 ((uint8_t) (0x01 << 1))
```

TIM3 Clock/trigger/slave mode selection [1] (in _TIM3_SMCR)

Definition at line 2041 of file STM8L10x.h.

5.2.2.740 _TIM3_SMS2

```
#define _TIM3_SMS2 ((uint8_t) (0x01 << 2))
```

TIM3 Clock/trigger/slave mode selection [2] (in _TIM3_SMCR)

Definition at line 2042 of file STM8L10x.h.

5.2.2.741 _TIM3_SR1

```
#define _TIM3_SR1 _SFR(uint8_t, TIM3_AddressBase+0x05)
```

TIM3 status register 1.

Definition at line 1978 of file STM8L10x.h.

5.2.2.742 _TIM3_SR1_RESET_VALUE

```
#define _TIM3_SR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 status register 1 reset value.

Definition at line 2002 of file STM8L10x.h.

5.2.2.743 _TIM3_SR2

```
#define _TIM3_SR2 _SFR(uint8_t, TIM3_AddressBase+0x06)
```

TIM3 status register 2.

Definition at line 1979 of file STM8L10x.h.

5.2.2.744 _TIM3_SR2_RESET_VALUE

```
#define _TIM3_SR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM3 status register 2 reset value.

Definition at line 2003 of file STM8L10x.h.

5.2.2.745 _TIM3_TG

```
#define _TIM3_TG ((uint8_t) (0x01 << 6))
```

TIM3 Trigger generation [0] (in _TIM3_EGR)

Definition at line 2089 of file STM8L10x.h.

5.2.2.746 _TIM3_TIE

```
#define _TIM3_TIE ((uint8_t) (0x01 << 6))
```

TIM3 Trigger interrupt enable [0] (in _TIM3_IER)

Definition at line 2067 of file STM8L10x.h.

5.2.2.747 _TIM3_TIF

```
#define _TIM3_TIF ((uint8_t) (0x01 << 6))
```

TIM3 Trigger interrupt flag [0] (in _TIM3_SR1)

Definition at line 2075 of file STM8L10x.h.

5.2.2.748 _TIM3_TS

```
#define _TIM3_TS ((uint8_t) (0x07 << 4))
```

TIM3 Trigger selection [2:0] (in _TIM3_SMCR)

Definition at line 2044 of file STM8L10x.h.

5.2.2.749 _TIM3_TS0

```
#define _TIM3_TS0 ((uint8_t) (0x01 << 4))
```

TIM3 Trigger selection [0] (in _TIM3_SMCR)

Definition at line 2045 of file STM8L10x.h.

5.2.2.750 _TIM3_TS1

```
#define _TIM3_TS1 ((uint8_t) (0x01 << 5))
```

TIM3 Trigger selection [1] (in _TIM3_SMCR)

Definition at line 2046 of file STM8L10x.h.

5.2.2.751 _TIM3_TS2

```
#define _TIM3_TS2 ((uint8_t) (0x01 << 6))
```

TIM3 Trigger selection [2] (in _TIM3_SMCR)

Definition at line 2047 of file STM8L10x.h.

5.2.2.752 _TIM3_UDIS

```
#define _TIM3_UDIS ((uint8_t) (0x01 << 1))
```

TIM3 Update disable [0] (in _TIM3_CR1)

Definition at line 2023 of file STM8L10x.h.

5.2.2.753 _TIM3_UG

```
#define _TIM3_UG ((uint8_t) (0x01 << 0))
```

TIM3 Update generation [0] (in _TIM3_EGR)

Definition at line 2085 of file STM8L10x.h.

5.2.2.754 _TIM3_UIE

```
#define _TIM3_UIE ((uint8_t) (0x01 << 0))
```

TIM3 Update interrupt enable [0] (in _TIM3_IER)

Definition at line 2063 of file STM8L10x.h.

5.2.2.755 _TIM3_UIF

```
#define _TIM3_UIF ((uint8_t) (0x01 << 0))
```

TIM3 Update interrupt flag [0] (in _TIM3_SR1)

Definition at line 2071 of file STM8L10x.h.

5.2.2.756 _TIM3_URS

```
#define _TIM3_URS ((uint8_t) (0x01 << 2))
```

TIM3 Update request source [0] (in _TIM3_CR1)

Definition at line 2024 of file STM8L10x.h.

5.2.2.757 _TIM4

```
#define _TIM4 _SFR(TIM4_t, TIM4_AddressBase)
```

TIM4 struct/bit access.

Definition at line 2259 of file STM8L10x.h.

5.2.2.758 _TIM4_ARPE

```
#define _TIM4_ARPE ((uint8_t) (0x01 << 7))
```

TIM4 Auto-reload preload enable [0] (in _TIM4_CR)

Definition at line 2287 of file STM8L10x.h.

5.2.2.759 _TIM4_ARR

```
#define _TIM4_ARR _SFR(uint8_t, TIM4_AddressBase+0x08)
```

TIM4 auto-reload register.

Definition at line 2268 of file STM8L10x.h.

5.2.2.760 _TIM4_ARR_RESET_VALUE

```
#define _TIM4_ARR_RESET_VALUE ((uint8_t) 0xFF)
```

TIM4 auto-reload register reset value.

Definition at line 2279 of file STM8L10x.h.

5.2.2.761 _TIM4_CEN

```
#define _TIM4_CEN ((uint8_t) (0x01 << 0))
```

TIM4 Counter enable [0] (in _TIM4_CR1)

Definition at line 2282 of file STM8L10x.h.

5.2.2.762 _TIM4_CNTR

```
#define _TIM4_CNTR _SFR(uint8_t, TIM4_AddressBase+0x06)
```

TIM4 counter register.

Definition at line 2266 of file STM8L10x.h.

5.2.2.763 _TIM4_CNTR_RESET_VALUE

```
#define _TIM4_CNTR_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 counter register reset value.

Definition at line 2277 of file STM8L10x.h.

5.2.2.764 _TIM4_CR1

```
#define _TIM4_CR1 _SFR(uint8_t, TIM4_AddressBase+0x00)
```

TIM4 control register 1.

Definition at line 2260 of file STM8L10x.h.

5.2.2.765 _TIM4_CR1_RESET_VALUE

```
#define _TIM4_CR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 control register 1 reset value.

Definition at line 2271 of file STM8L10x.h.

5.2.2.766 _TIM4_CR2

```
#define _TIM4_CR2 _SFR(uint8_t, TIM4_AddressBase+0x01)
```

TIM4 control register 2.

Definition at line 2261 of file STM8L10x.h.

5.2.2.767 _TIM4_CR2_RESET_VALUE

```
#define _TIM4_CR2_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 control register 2 reset value.

Definition at line 2272 of file STM8L10x.h.

5.2.2.768 _TIM4_EGR

```
#define _TIM4_EGR _SFR(uint8_t, TIM4_AddressBase+0x05)
```

TIM4 event generation register.

Definition at line 2265 of file STM8L10x.h.

5.2.2.769 _TIM4_EGR_RESET_VALUE

```
#define _TIM4_EGR_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 event generation register reset value.

Definition at line 2276 of file STM8L10x.h.

5.2.2.770 _TIM4_IER

```
#define _TIM4_IER _SFR(uint8_t, TIM4_AddressBase+0x03)
```

TIM4 interrupt enable register.

Definition at line 2263 of file STM8L10x.h.

5.2.2.771 _TIM4_IER_RESET_VALUE

```
#define _TIM4_IER_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 interrupt enable register reset value.

Definition at line 2274 of file STM8L10x.h.

5.2.2.772 _TIM4_MMS

```
#define _TIM4_MMS ((uint8_t) (0x07 << 4))
```

TIM4 Master mode selection [2:0] (in _TIM4_CR2)

Definition at line 2291 of file STM8L10x.h.

5.2.2.773 _TIM4_MMS0

```
#define _TIM4_MMS0 ((uint8_t) (0x01 << 4))
```

TIM4 Master mode selection [0] (in _TIM4_CR2)

Definition at line 2292 of file STM8L10x.h.

5.2.2.774 _TIM4_MMS1

```
#define _TIM4_MMS1 ((uint8_t) (0x01 << 5))
```

TIM4 Master mode selection [1] (in _TIM4_CR2)

Definition at line 2293 of file STM8L10x.h.

5.2.2.775 _TIM4_MMS2

```
#define _TIM4_MMS2 ((uint8_t) (0x01 << 6))
```

TIM4 Master mode selection [2] (in _TIM4_CR2)

Definition at line 2294 of file STM8L10x.h.

5.2.2.776 _TIM4_MSM

```
#define _TIM4_MSM ((uint8_t) (0x01 << 7))
```

TIM4 Master/slave mode [0] (in _TIM4_SMCR)

Definition at line 2307 of file STM8L10x.h.

5.2.2.777 _TIM4_OPM

```
#define _TIM4_OPM ((uint8_t) (0x01 << 3))
```

TIM4 One-pulse mode [0] (in _TIM4_CR1)

Definition at line 2285 of file STM8L10x.h.

5.2.2.778 _TIM4_PSC

```
#define _TIM4_PSC ((uint8_t) (0x0F << 0))
```

TIM4 clock prescaler [3:0] (in _TIM4_PSCR)

Definition at line 2328 of file STM8L10x.h.

5.2.2.779 _TIM4_PSC0

```
#define _TIM4_PSC0 ((uint8_t) (0x01 << 0))
```

TIM4 clock prescaler [0] (in _TIM4_PSCR)

Definition at line 2329 of file STM8L10x.h.

5.2.2.780 _TIM4_PSC1

```
#define _TIM4_PSC1 ((uint8_t) (0x01 << 1))
```

TIM4 clock prescaler [1] (in _TIM4_PSCR)

Definition at line 2330 of file STM8L10x.h.

5.2.2.781 _TIM4_PSC2

```
#define _TIM4_PSC2 ((uint8_t) (0x01 << 2))
```

TIM4 clock prescaler [2] (in _TIM4_PSCR)

Definition at line 2331 of file STM8L10x.h.

5.2.2.782 _TIM4_PSC3

```
#define _TIM4_PSC3 ((uint8_t) (0x01 << 3))
```

TIM4 clock prescaler [3] (in _TIM4_PSCR)

Definition at line 2332 of file STM8L10x.h.

5.2.2.783 _TIM4_PSCR

```
#define _TIM4_PSCR _SFR(uint8_t, TIM4_AddressBase+0x07)
```

TIM4 clock prescaler register.

Definition at line 2267 of file STM8L10x.h.

5.2.2.784 _TIM4_PSCR_RESET_VALUE

```
#define _TIM4_PSCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 clock prescaler register reset value.

Definition at line 2278 of file STM8L10x.h.

5.2.2.785 _TIM4_SMCR

```
#define _TIM4_SMCR _SFR(uint8_t, TIM4_AddressBase+0x02)
```

TIM4 Slave mode control register.

Definition at line 2262 of file STM8L10x.h.

5.2.2.786 _TIM4_SMCR_RESET_VALUE

```
#define _TIM4_SMCR_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 Slave mode control register reset value.

Definition at line 2273 of file STM8L10x.h.

5.2.2.787 _TIM4_SMS

```
#define _TIM4_SMS ((uint8_t) (0x07 << 0))
```

TIM4 Clock/trigger/slave mode selection [2:0] (in _TIM4_SMCR)

Definition at line 2298 of file STM8L10x.h.

5.2.2.788 _TIM4_SMS0

```
#define _TIM4_SMS0 ((uint8_t) (0x01 << 0))
```

TIM4 Clock/trigger/slave mode selection [0] (in _TIM4_SMCR)

Definition at line 2299 of file STM8L10x.h.

5.2.2.789 _TIM4_SMS1

```
#define _TIM4_SMS1 ((uint8_t) (0x01 << 1))
```

TIM4 Clock/trigger/slave mode selection [1] (in _TIM4_SMCR)

Definition at line 2300 of file STM8L10x.h.

5.2.2.790 _TIM4_SMS2

```
#define _TIM4_SMS2 ((uint8_t) (0x01 << 2))
```

TIM4 Clock/trigger/slave mode selection [2] (in _TIM4_SMCR)

Definition at line 2301 of file STM8L10x.h.

5.2.2.791 _TIM4_SR1

```
#define _TIM4_SR1 _SFR(uint8_t, TIM4_AddressBase+0x04)
```

TIM4 status register.

Definition at line 2264 of file STM8L10x.h.

5.2.2.792 _TIM4_SR1_RESET_VALUE

```
#define _TIM4_SR1_RESET_VALUE ((uint8_t) 0x00)
```

TIM4 status register reset value.

Definition at line 2275 of file STM8L10x.h.

5.2.2.793 _TIM4_TG

```
#define _TIM4_TG ((uint8_t) (0x01 << 6))
```

TIM4 Trigger generation [0] (in _TIM4_EGR)

Definition at line 2324 of file STM8L10x.h.

5.2.2.794 _TIM4_TIE

```
#define _TIM4_TIE ((uint8_t) (0x01 << 6))
```

TIM4 Trigger interrupt enable [0] (in _TIM4_IER)

Definition at line 2312 of file STM8L10x.h.

5.2.2.795 _TIM4_TIF

```
#define _TIM4_TIF ((uint8_t) (0x01 << 6))
```

TIM4 Trigger interrupt flag [0] (in _TIM4_SR1)

Definition at line 2318 of file STM8L10x.h.

5.2.2.796 _TIM4_TS

```
#define _TIM4_TS ((uint8_t) (0x07 << 4))
```

TIM4 Trigger selection [2:0] (in _TIM4_SMCR)

Definition at line 2303 of file STM8L10x.h.

5.2.2.797 _TIM4_TS0

```
#define _TIM4_TS0 ((uint8_t) (0x01 << 4))
```

TIM4 Trigger selection [0] (in _TIM4_SMCR)

Definition at line 2304 of file STM8L10x.h.

5.2.2.798 _TIM4_TS1

```
#define _TIM4_TS1 ((uint8_t) (0x01 << 5))
```

TIM4 Trigger selection [1] (in _TIM4_SMCR)

Definition at line 2305 of file STM8L10x.h.

5.2.2.799 _TIM4_TS2

```
#define _TIM4_TS2 ((uint8_t) (0x01 << 6))
```

TIM4 Trigger selection [2] (in _TIM4_SMCR)

Definition at line 2306 of file STM8L10x.h.

5.2.2.800 _TIM4_UDIS

```
#define _TIM4_UDIS ((uint8_t) (0x01 << 1))
```

TIM4 Update disable [0] (in _TIM4_CR1)

Definition at line 2283 of file STM8L10x.h.

5.2.2.801 _TIM4_UG

```
#define _TIM4_UG ((uint8_t) (0x01 << 0))
```

TIM4 Update generation [0] (in _TIM4_EGR)

Definition at line 2322 of file STM8L10x.h.

5.2.2.802 _TIM4_UIE

```
#define _TIM4_UIE ((uint8_t) (0x01 << 0))
```

TIM4 Update interrupt enable [0] (in _TIM4_IER)

Definition at line 2310 of file STM8L10x.h.

5.2.2.803 _TIM4_UIF

```
#define _TIM4_UIF ((uint8_t) (0x01 << 0))
```

TIM4 Update interrupt flag [0] (in _TIM4_SR1)

Definition at line 2316 of file STM8L10x.h.

5.2.2.804 _TIM4_URS

```
#define _TIM4_URS ((uint8_t) (0x01 << 2))
```

TIM4 Update request source [0] (in _TIM4_CR1)

Definition at line 2284 of file STM8L10x.h.

5.2.2.805 _USART

```
#define _USART _SFR(USART_t, USART_AddressBase)
```

USART struct/bit access.

Definition at line 1412 of file STM8L10x.h.

5.2.2.806 _USART_ADD

```
#define _USART_ADD ((uint8_t) (0x0F << 0))
```

USART Address of the UART node [3:0] (in _USART_CR4)

Definition at line 1472 of file STM8L10x.h.

5.2.2.807 _USART_ADD0

```
#define _USART_ADD0 ((uint8_t) (0x01 << 0))
```

USART Address of the UART node [0] (in _USART_CR4)

Definition at line 1473 of file STM8L10x.h.

5.2.2.808 _USART_ADD1

```
#define _USART_ADD1 ((uint8_t) (0x01 << 1))
```

USART Address of the UART node [1] (in _USART_CR4)

Definition at line 1474 of file STM8L10x.h.

5.2.2.809 _USART_ADD2

```
#define _USART_ADD2 ((uint8_t) (0x01 << 2))
```

USART Address of the UART node [2] (in _USART_CR4)

Definition at line 1475 of file STM8L10x.h.

5.2.2.810 _USART_ADD3

```
#define _USART_ADD3 ((uint8_t) (0x01 << 3))
```

USART Address of the UART node [3] (in _USART_CR4)

Definition at line 1476 of file STM8L10x.h.

5.2.2.811 _USART_BRR1

```
#define _USART_BRR1 _SFR(uint8_t, USART_AddressBase+0x02)
```

USART Baud rate register 1.

Definition at line 1415 of file STM8L10x.h.

5.2.2.812 _USART_BRR1_RESET_VALUE

```
#define _USART_BRR1_RESET_VALUE ((uint8_t) 0x00)
```

USART Baud rate register 1 reset value.

Definition at line 1424 of file STM8L10x.h.

5.2.2.813 _USART_BRR2

```
#define _USART_BRR2 _SFR(uint8_t, USART_AddressBase+0x03)
```

USART Baud rate register 2.

Definition at line 1416 of file STM8L10x.h.

5.2.2.814 _USART_BRR2_RESET_VALUE

```
#define _USART_BRR2_RESET_VALUE ((uint8_t) 0x00)
```

USART Baud rate register 2 reset value.

Definition at line 1425 of file STM8L10x.h.

5.2.2.815 _USART_CKEN

```
#define _USART_CKEN ((uint8_t) (0x01 << 3))
```

USART Clock enable [0] (in _USART_CR3)

Definition at line 1465 of file STM8L10x.h.

5.2.2.816 _USART_CPHA

```
#define _USART_CPHA ((uint8_t) (0x01 << 1))
```

USART Clock phase [0] (in _USART_CR3)

Definition at line 1463 of file STM8L10x.h.

5.2.2.817 _USART_CPOL

```
#define _USART_CPOL ((uint8_t) (0x01 << 2))
```

USART Clock polarity [0] (in _USART_CR3)

Definition at line 1464 of file STM8L10x.h.

5.2.2.818 _USART_CR1

```
#define _USART_CR1 _SFR(uint8_t, USART_AddressBase+0x04)
```

USART Control register 1.

Definition at line 1417 of file STM8L10x.h.

5.2.2.819 _USART_CR1_RESET_VALUE

```
#define _USART_CR1_RESET_VALUE ((uint8_t) 0x00)
```

USART Control register 1 reset value.

Definition at line 1426 of file STM8L10x.h.

5.2.2.820 _USART_CR2

```
#define _USART_CR2 _SFR(uint8_t, USART_AddressBase+0x05)
```

USART Control register 2.

Definition at line 1418 of file STM8L10x.h.

5.2.2.821 _USART_CR2_RESET_VALUE

```
#define _USART_CR2_RESET_VALUE ((uint8_t) 0x00)
```

USART Control register 2 reset value.

Definition at line 1427 of file STM8L10x.h.

5.2.2.822 _USART_CR3

```
#define _USART_CR3 _SFR(uint8_t, USART_AddressBase+0x06)
```

USART Control register 3.

Definition at line 1419 of file STM8L10x.h.

5.2.2.823 _USART_CR3_RESET_VALUE

```
#define _USART_CR3_RESET_VALUE ((uint8_t) 0x00)
```

USART Control register 3 reset value.

Definition at line 1428 of file STM8L10x.h.

5.2.2.824 _USART_CR4

```
#define _USART_CR4 _SFR(uint8_t, USART_AddressBase+0x07)
```

USART Control register 4.

Definition at line 1420 of file STM8L10x.h.

5.2.2.825 _USART_CR4_RESET_VALUE

```
#define _USART_CR4_RESET_VALUE ((uint8_t) 0x00)
```

USART Control register 4 reset value.

Definition at line 1429 of file STM8L10x.h.

5.2.2.826 _USART_DR

```
#define _USART_DR _SFR(uint8_t, USART_AddressBase+0x01)
```

USART data register.

Definition at line 1414 of file STM8L10x.h.

5.2.2.827 _USART_FE

```
#define _USART_FE ((uint8_t) (0x01 << 1))
```

USART Framing error [0] (in _USART_SR)

Definition at line 1433 of file STM8L10x.h.

5.2.2.828 _USART_IDLE

```
#define _USART_IDLE ((uint8_t) (0x01 << 4))
```

USART IDLE line detected [0] (in _USART_SR)

Definition at line 1436 of file STM8L10x.h.

5.2.2.829 _USART_ILIEN

```
#define _USART_ILIEN ((uint8_t) (0x01 << 4))
```

USART IDLE Line interrupt enable [0] (in _USART_CR2)

Definition at line 1456 of file STM8L10x.h.

5.2.2.830 _USART_LBCL

```
#define _USART_LBCL ((uint8_t) (0x01 << 0))
```

USART Last bit clock pulse [0] (in _USART_CR3)

Definition at line 1462 of file STM8L10x.h.

5.2.2.831 _USART_M

```
#define _USART_M ((uint8_t) (0x01 << 4))
```

USART word length [0] (in _USART_CR1)

Definition at line 1446 of file STM8L10x.h.

5.2.2.832 _USART_NF

```
#define _USART_NF ((uint8_t) (0x01 << 2))
```

USART Noise flag [0] (in _USART_SR)

Definition at line 1434 of file STM8L10x.h.

5.2.2.833 _USART_OR_LHE

```
#define _USART_OR_LHE ((uint8_t) (0x01 << 3))
```

USART LIN Header Error (LIN Slave mode) / Overrun error [0] (in _USART_SR)

Definition at line 1435 of file STM8L10x.h.

5.2.2.834 _USART_PCEN

```
#define _USART_PCEN ((uint8_t) (0x01 << 2))
```

USART Parity control enable [0] (in _USART_CR1)

Definition at line 1444 of file STM8L10x.h.

5.2.2.835 _USART_PE

```
#define _USART_PE ((uint8_t) (0x01 << 0))
```

USART Parity error [0] (in _USART_SR)

Definition at line 1432 of file STM8L10x.h.

5.2.2.836 _USART_PIEN

```
#define _USART_PIEN ((uint8_t) (0x01 << 0))
```

USART Parity interrupt enable [0] (in _USART_CR1)

Definition at line 1442 of file STM8L10x.h.

5.2.2.837 _USART_PS

```
#define _USART_PS ((uint8_t) (0x01 << 1))
```

USART Parity selection [0] (in _USART_CR1)

Definition at line 1443 of file STM8L10x.h.

5.2.2.838 _USART_R8

```
#define _USART_R8 ((uint8_t) (0x01 << 7))
```

USART Receive Data bit 8 (in 9-bit mode) [0] (in _USART_CR1)

Definition at line 1449 of file STM8L10x.h.

5.2.2.839 _USART_REN

```
#define _USART_REN ((uint8_t) (0x01 << 2))
```

USART Receiver enable [0] (in _USART_CR2)

Definition at line 1454 of file STM8L10x.h.

5.2.2.840 _USART_RIEN

```
#define _USART_RIEN ((uint8_t) (0x01 << 5))
```

USART Receiver interrupt enable [0] (in _USART_CR2)

Definition at line 1457 of file STM8L10x.h.

5.2.2.841 _USART_RWU

```
#define _USART_RWU ((uint8_t) (0x01 << 1))
```

USART Receiver wakeup [0] (in _USART_CR2)

Definition at line 1453 of file STM8L10x.h.

5.2.2.842 _USART_RXNE

```
#define _USART_RXNE ((uint8_t) (0x01 << 5))
```

USART Read data register not empty [0] (in _USART_SR)

Definition at line 1437 of file STM8L10x.h.

5.2.2.843 _USART_SBK

```
#define _USART_SBK ((uint8_t) (0x01 << 0))
```

USART Send break [0] (in _USART_CR2)

Definition at line 1452 of file STM8L10x.h.

5.2.2.844 _USART_SR

```
#define _USART_SR _SFR(uint8_t, USART_AddressBase+0x00)
```

USART Status register.

Definition at line 1413 of file STM8L10x.h.

5.2.2.845 _USART_SR_RESET_VALUE

```
#define _USART_SR_RESET_VALUE ((uint8_t) 0xC0)
```

USART Status register reset value.

Definition at line 1423 of file STM8L10x.h.

5.2.2.846 _USART_STOP

```
#define _USART_STOP ((uint8_t) (0x03 << 4))
```

USART STOP bits [1:0] (in _USART_CR3)

Definition at line 1466 of file STM8L10x.h.

5.2.2.847 _USART_STOP0

```
#define _USART_STOP0 ((uint8_t) (0x01 << 4))
```

USART STOP bits [0] (in _USART_CR3)

Definition at line 1467 of file STM8L10x.h.

5.2.2.848 _USART_STOP1

```
#define _USART_STOP1 ((uint8_t) (0x01 << 5))
```

USART STOP bits [1] (in _USART_CR3)

Definition at line 1468 of file STM8L10x.h.

5.2.2.849 _USART_T8

```
#define _USART_T8 ((uint8_t) (0x01 << 6))
```

USART Transmit Data bit 8 (in 9-bit mode) [0] (in _USART_CR1)

Definition at line 1448 of file STM8L10x.h.

5.2.2.850 _USART_TC

```
#define _USART_TC ((uint8_t) (0x01 << 6))
```

USART Transmission complete [0] (in _USART_SR)

Definition at line 1438 of file STM8L10x.h.

5.2.2.851 _USART_TCIEN

```
#define _USART_TCIEN ((uint8_t) (0x01 << 6))
```

USART Transmission complete interrupt enable [0] (in _USART_CR2)

Definition at line 1458 of file STM8L10x.h.

5.2.2.852 _USART_TEN

```
#define _USART_TEN ((uint8_t) (0x01 << 3))
```

USART Transmitter enable [0] (in _USART_CR2)

Definition at line 1455 of file STM8L10x.h.

5.2.2.853 _USART_TIEN

```
#define _USART_TIEN ((uint8_t) (0x01 << 7))
```

USART Transmitter interrupt enable [0] (in _USART_CR2)

Definition at line 1459 of file STM8L10x.h.

5.2.2.854 _USART_TXE

```
#define _USART_TXE ((uint8_t) (0x01 << 7))
```

USART Transmit data register empty [0] (in _USART_SR)

Definition at line 1439 of file STM8L10x.h.

5.2.2.855 _USART_UARTD

```
#define _USART_UARTD ((uint8_t) (0x01 << 5))
```

USART Disable (for low power consumption) [0] (in _USART_CR1)

Definition at line 1447 of file STM8L10x.h.

5.2.2.856 _USART_WAKE

```
#define _USART_WAKE ((uint8_t) (0x01 << 3))
```

USART Wakeup method [0] (in _USART_CR1)

Definition at line 1445 of file STM8L10x.h.

5.2.2.857 _WFE

```
#define _WFE _SFR(WFE_t, WFE_AddressBase)
```

WFE struct/bit access.

Definition at line 1517 of file STM8L10x.h.

5.2.2.858 _WFE_CR1

```
#define _WFE_CR1 _SFR(uint8_t, WFE_AddressBase+0x00)
```

WFE Control register 1.

Definition at line 1518 of file STM8L10x.h.

5.2.2.859 _WFE_CR1_RESET_VALUE

```
#define _WFE_CR1_RESET_VALUE ((uint8_t) 0x03)
```

WFE Control register 1 reset value.

Definition at line 1522 of file STM8L10x.h.

5.2.2.860 _WFE_CR2

```
#define _WFE_CR2 _SFR(uint8_t, WFE_AddressBase+0x01)
```

WFE Control register 2.

Definition at line 1519 of file STM8L10x.h.

5.2.2.861 _WFE_CR2_RESET_VALUE

```
#define _WFE_CR2_RESET_VALUE ((uint8_t) 0x00)
```

WFE Control register 2 reset value.

Definition at line 1523 of file STM8L10x.h.

5.2.2.862 _WFE_EXTI_EV0

```
#define _WFE_EXTI_EV0 ((uint8_t) (0x01 << 4))
```

Interrupt on pin 0 of all ports event [0] (in _WFE_CR1)

Definition at line 1529 of file STM8L10x.h.

5.2.2.863 _WFE_EXTI_EV1

```
#define _WFE_EXTI_EV1 ((uint8_t) (0x01 << 5))
```

Interrupt on pin 1 of all ports event [0] (in _WFE_CR1)

Definition at line 1530 of file STM8L10x.h.

5.2.2.864 _WFE_EXTI_EV2

```
#define _WFE_EXTI_EV2 ((uint8_t) (0x01 << 6))
```

Interrupt on pin 2 of all ports event [0] (in _WFE_CR1)

Definition at line 1531 of file STM8L10x.h.

5.2.2.865 _WFE_EXTI_EV3

```
#define _WFE_EXTI_EV3 ((uint8_t) (0x01 << 7))
```

Interrupt on pin 3 of all ports event [0] (in _WFE_CR1)

Definition at line 1532 of file STM8L10x.h.

5.2.2.866 _WFE_EXTI_EV4

```
#define _WFE_EXTI_EV4 ((uint8_t) (0x01 << 0))
```

Interrupt on pin 4 of all ports event [0] (in _WFE_CR1)

Definition at line 1535 of file STM8L10x.h.

5.2.2.867 _WFE_EXTI_EV5

```
#define _WFE_EXTI_EV5 ((uint8_t) (0x01 << 1))
```

Interrupt on pin 5 of all ports event [0] (in _WFE_CR1)

Definition at line 1536 of file STM8L10x.h.

5.2.2.868 _WFE_EXTI_EV6

```
#define _WFE_EXTI_EV6 ((uint8_t) (0x01 << 2))
```

Interrupt on pin 6 of all ports event [0] (in _WFE_CR1)

Definition at line 1537 of file STM8L10x.h.

5.2.2.869 _WFE_EXTI_EV7

```
#define _WFE_EXTI_EV7 ((uint8_t) (0x01 << 3))
```

Interrupt on pin 7 of all ports event [0] (in _WFE_CR1)

Definition at line 1538 of file STM8L10x.h.

5.2.2.870 _WFE_EXTI_EVB

```
#define _WFE_EXTI_EVB ((uint8_t) (0x01 << 4))
```

Interrupt on port B event [0] (in _WFE_CR1)

Definition at line 1539 of file STM8L10x.h.

5.2.2.871 _WFE_EXTI_EVD

```
#define _WFE_EXTI_EVD ((uint8_t) (0x01 << 5))
```

Interrupt on port D event [0] (in _WFE_CR1)

Definition at line 1540 of file STM8L10x.h.

5.2.2.872 _WFE_TIM2_EV0

```
#define _WFE_TIM2_EV0 ((uint8_t) (0x01 << 0))
```

TIM2 update, trigger or break event [0] (in _WFE_CR1)

Definition at line 1526 of file STM8L10x.h.

5.2.2.873 _WFE_TIM2_EV1

```
#define _WFE_TIM2_EV1 ((uint8_t) (0x01 << 1))
```

TIM2 capture or compare event [0] (in _WFE_CR1)

Definition at line 1527 of file STM8L10x.h.

5.2.2.874 AWU_AddressBase [1/6]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 62 of file STM8L101F2.h.

5.2.2.875 AWU_AddressBase [2/6]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 62 of file STM8L101K3.h.

5.2.2.876 AWU_AddressBase [3/6]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 62 of file STM8L101G3.h.

5.2.2.877 AWU_AddressBase [4/6]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 62 of file STM8L101F3.h.

5.2.2.878 AWU_AddressBase [5/6]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 62 of file STM8L101F1.h.

5.2.2.879 AWU_AddressBase [6/6]

```
#define AWU_AddressBase 0x50F0
```

Definition at line 62 of file STM8L101G2.h.

5.2.2.880 BEEP_AddressBase [1/6]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 63 of file STM8L101G3.h.

5.2.2.881 BEEP_AddressBase [2/6]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 63 of file STM8L101F1.h.

5.2.2.882 BEEP_AddressBase [3/6]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 63 of file STM8L101K3.h.

5.2.2.883 BEEP_AddressBase [4/6]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 63 of file STM8L101F2.h.

5.2.2.884 BEEP_AddressBase [5/6]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 63 of file STM8L101F3.h.

5.2.2.885 BEEP_AddressBase [6/6]

```
#define BEEP_AddressBase 0x50F3
```

Definition at line 63 of file STM8L101G2.h.

5.2.2.886 CFG_AddressBase [1/6]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 73 of file STM8L101F2.h.

5.2.2.887 CFG_AddressBase [2/6]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 73 of file STM8L101G2.h.

5.2.2.888 CFG_AddressBase [3/6]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 73 of file STM8L101F3.h.

5.2.2.889 CFG_AddressBase [4/6]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 73 of file STM8L101K3.h.

5.2.2.890 CFG_AddressBase [5/6]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 73 of file STM8L101G3.h.

5.2.2.891 CFG_AddressBase [6/6]

```
#define CFG_AddressBase 0x7F60
```

Definition at line 73 of file STM8L101F1.h.

5.2.2.892 CLK_AddressBase [1/6]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 60 of file STM8L101F2.h.

5.2.2.893 CLK_AddressBase [2/6]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 60 of file STM8L101G3.h.

5.2.2.894 CLK_AddressBase [3/6]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 60 of file STM8L101K3.h.

5.2.2.895 CLK_AddressBase [4/6]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 60 of file STM8L101F3.h.

5.2.2.896 CLK_AddressBase [5/6]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 60 of file STM8L101F1.h.

5.2.2.897 CLK_AddressBase [6/6]

```
#define CLK_AddressBase 0x50C0
```

Definition at line 60 of file STM8L101G2.h.

5.2.2.898 COMP_AddressBase [1/6]

```
#define COMP_AddressBase 0x5300
```

Definition at line 72 of file STM8L101K3.h.

5.2.2.899 COMP_AddressBase [2/6]

```
#define COMP_AddressBase 0x5300
```

Definition at line 72 of file STM8L101G3.h.

5.2.2.900 COMP_AddressBase [3/6]

```
#define COMP_AddressBase 0x5300
```

Definition at line 72 of file STM8L101F3.h.

5.2.2.901 COMP_AddressBase [4/6]

```
#define COMP_AddressBase 0x5300
```

Definition at line 72 of file STM8L101F1.h.

5.2.2.902 COMP_AddressBase [5/6]

```
#define COMP_AddressBase 0x5300
```

Definition at line 72 of file STM8L101G2.h.

5.2.2.903 COMP_AddressBase [6/6]

```
#define COMP_AddressBase 0x5300
```

Definition at line 72 of file STM8L101F2.h.

5.2.2.904 DISABLE_INTERRUPTS

```
#define DISABLE_INTERRUPTS( ) __asm__("sim")
```

disable interrupt handling

Definition at line 169 of file STM8L10x.h.

5.2.2.905 DM_AddressBase [1/6]

```
#define DM_AddressBase 0x7F90
```

Definition at line 75 of file STM8L101G3.h.

5.2.2.906 DM_AddressBase [2/6]

```
#define DM_AddressBase 0x7F90
```

Definition at line 75 of file STM8L101G2.h.

5.2.2.907 DM_AddressBase [3/6]

```
#define DM_AddressBase 0x7F90
```

Definition at line 75 of file STM8L101F2.h.

5.2.2.908 DM_AddressBase [4/6]

```
#define DM_AddressBase 0x7F90
```

Definition at line 75 of file STM8L101K3.h.

5.2.2.909 DM_AddressBase [5/6]

```
#define DM_AddressBase 0x7F90
```

Definition at line 75 of file STM8L101F1.h.

5.2.2.910 DM_AddressBase [6/6]

```
#define DM_AddressBase 0x7F90
```

Definition at line 75 of file STM8L101F3.h.

5.2.2.911 ENABLE_INTERRUPTS

```
#define ENABLE_INTERRUPTS( ) __asm__("rim")
```

enable interrupt handling

Definition at line 170 of file STM8L10x.h.

5.2.2.912 ENTER_HALT

```
#define ENTER_HALT( ) __asm__("halt")
```

put controller to HALT mode

Definition at line 173 of file STM8L10x.h.

5.2.2.913 EXTI_AddressBase [1/6]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 58 of file STM8L101G2.h.

5.2.2.914 EXTI_AddressBase [2/6]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 58 of file STM8L101G3.h.

5.2.2.915 EXTI_AddressBase [3/6]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 58 of file STM8L101K3.h.

5.2.2.916 EXTI_AddressBase [4/6]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 58 of file STM8L101F3.h.

5.2.2.917 EXTI_AddressBase [5/6]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 58 of file STM8L101F1.h.

5.2.2.918 EXTI_AddressBase [6/6]

```
#define EXTI_AddressBase 0x50A0
```

Definition at line 58 of file STM8L101F2.h.

5.2.2.919 FLASH_AddressBase [1/6]

```
#define FLASH_AddressBase 0x5050
```

Definition at line 57 of file STM8L101F1.h.

5.2.2.920 FLASH_AddressBase [2/6]

```
#define FLASH_AddressBase 0x5050
```

Definition at line 57 of file STM8L101F3.h.

5.2.2.921 FLASH_AddressBase [3/6]

```
#define FLASH_AddressBase 0x5050
```

Definition at line 57 of file STM8L101G3.h.

5.2.2.922 FLASH_AddressBase [4/6]

```
#define FLASH_AddressBase 0x5050
```

Definition at line 57 of file STM8L101G2.h.

5.2.2.923 FLASH_AddressBase [5/6]

```
#define FLASH_AddressBase 0x5050
```

Definition at line 57 of file STM8L101F2.h.

5.2.2.924 FLASH_AddressBase [6/6]

```
#define FLASH_AddressBase 0x5050
```

Definition at line 57 of file STM8L101K3.h.

5.2.2.925 I2C_AddressBase [1/6]

```
#define I2C_AddressBase 0x5210
```

Definition at line 65 of file STM8L101F3.h.

5.2.2.926 I2C_AddressBase [2/6]

```
#define I2C_AddressBase 0x5210
```

Definition at line 65 of file STM8L101G2.h.

5.2.2.927 I2C_AddressBase [3/6]

```
#define I2C_AddressBase 0x5210
```

Definition at line 65 of file STM8L101F2.h.

5.2.2.928 I2C_AddressBase [4/6]

```
#define I2C_AddressBase 0x5210
```

Definition at line 65 of file STM8L101K3.h.

5.2.2.929 I2C_AddressBase [5/6]

```
#define I2C_AddressBase 0x5210
```

Definition at line 65 of file STM8L101F1.h.

5.2.2.930 I2C_AddressBase [6/6]

```
#define I2C_AddressBase 0x5210
```

Definition at line 65 of file STM8L101G3.h.

5.2.2.931 IRTIM_AddressBase [1/6]

```
#define IRTIM_AddressBase 0x52FF
```

Definition at line 71 of file STM8L101G3.h.

5.2.2.932 IRTIM_AddressBase [2/6]

```
#define IRTIM_AddressBase 0x52FF
```

Definition at line 71 of file STM8L101G2.h.

5.2.2.933 IRTIM_AddressBase [3/6]

```
#define IRTIM_AddressBase 0x52FF
```

Definition at line 71 of file STM8L101F1.h.

5.2.2.934 IRTIM_AddressBase [4/6]

```
#define IRTIM_AddressBase 0x52FF
```

Definition at line 71 of file STM8L101K3.h.

5.2.2.935 IRTIM_AddressBase [5/6]

```
#define IRTIM_AddressBase 0x52FF
```

Definition at line 71 of file STM8L101F3.h.

5.2.2.936 IRTIM_AddressBase [6/6]

```
#define IRTIM_AddressBase 0x52FF
```

Definition at line 71 of file STM8L101F2.h.

5.2.2.937 ISR_HANDLER

```
#define ISR_HANDLER(
              func,
              irq ) void func(void) __interrupt(irq)
```

handler for interrupt service routine

Definition at line 160 of file STM8L10x.h.

5.2.2.938 ISR_HANDLER_TRAP

```
#define ISR_HANDLER_TRAP(
              func ) void func() __trap
```

handler for trap service routine

Definition at line 162 of file STM8L10x.h.

5.2.2.939 ITC_AddressBase [1/6]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 74 of file STM8L101F1.h.

5.2.2.940 ITC_AddressBase [2/6]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 74 of file STM8L101F2.h.

5.2.2.941 ITC_AddressBase [3/6]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 74 of file STM8L101G2.h.

5.2.2.942 ITC_AddressBase [4/6]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 74 of file STM8L101K3.h.

5.2.2.943 ITC_AddressBase [5/6]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 74 of file STM8L101F3.h.

5.2.2.944 ITC_AddressBase [6/6]

```
#define ITC_AddressBase 0x7F70
```

Definition at line 74 of file STM8L101G3.h.

5.2.2.945 IWDG_AddressBase [1/6]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 61 of file STM8L101G3.h.

5.2.2.946 IWDG_AddressBase [2/6]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 61 of file STM8L101F3.h.

5.2.2.947 IWDG_AddressBase [3/6]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 61 of file STM8L101K3.h.

5.2.2.948 IWDG_AddressBase [4/6]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 61 of file STM8L101F1.h.

5.2.2.949 IWDG_AddressBase [5/6]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 61 of file STM8L101F2.h.

5.2.2.950 IWDG_AddressBase [6/6]

```
#define IWDG_AddressBase 0x50E0
```

Definition at line 61 of file STM8L101G2.h.

5.2.2.951 NOP

```
#define NOP( ) __asm__("nop")
```

perform a nop() operation (=minimum delay)

Definition at line 168 of file STM8L10x.h.

5.2.2.952 OPT_AddressBase [1/6]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8L101F1.h.

5.2.2.953 OPT_AddressBase [2/6]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8L101F3.h.

5.2.2.954 OPT_AddressBase [3/6]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8L101K3.h.

5.2.2.955 OPT_AddressBase [4/6]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8L101G3.h.

5.2.2.956 OPT_AddressBase [5/6]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8L101F2.h.

5.2.2.957 OPT_AddressBase [6/6]

```
#define OPT_AddressBase 0x4800
```

Definition at line 52 of file STM8L101G2.h.

5.2.2.958 PORTA_AddressBase [1/6]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8L101F3.h.

5.2.2.959 PORTA_AddressBase [2/6]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8L101F2.h.

5.2.2.960 PORTA_AddressBase [3/6]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8L101K3.h.

5.2.2.961 PORTA_AddressBase [4/6]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8L101G2.h.

5.2.2.962 PORTA_AddressBase [5/6]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8L101F1.h.

5.2.2.963 PORTA_AddressBase [6/6]

```
#define PORTA_AddressBase 0x5000
```

Definition at line 53 of file STM8L101G3.h.

5.2.2.964 PORTB_AddressBase [1/6]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8L101K3.h.

5.2.2.965 PORTB_AddressBase [2/6]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8L101F2.h.

5.2.2.966 PORTB_AddressBase [3/6]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8L101F1.h.

5.2.2.967 PORTB_AddressBase [4/6]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8L101F3.h.

5.2.2.968 PORTB_AddressBase [5/6]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8L101G3.h.

5.2.2.969 PORTB_AddressBase [6/6]

```
#define PORTB_AddressBase 0x5005
```

Definition at line 54 of file STM8L101G2.h.

5.2.2.970 PORTC_AddressBase [1/6]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8L101K3.h.

5.2.2.971 PORTC_AddressBase [2/6]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8L101F1.h.

5.2.2.972 PORTC_AddressBase [3/6]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8L101F2.h.

5.2.2.973 PORTC_AddressBase [4/6]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8L101F3.h.

5.2.2.974 PORTC_AddressBase [5/6]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8L101G3.h.

5.2.2.975 PORTC_AddressBase [6/6]

```
#define PORTC_AddressBase 0x500A
```

Definition at line 55 of file STM8L101G2.h.

5.2.2.976 PORTD_AddressBase [1/6]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8L101G2.h.

5.2.2.977 PORTD_AddressBase [2/6]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8L101F1.h.

5.2.2.978 PORTD_AddressBase [3/6]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8L101F2.h.

5.2.2.979 PORTD_AddressBase [4/6]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8L101G3.h.

5.2.2.980 PORTD_AddressBase [5/6]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8L101K3.h.

5.2.2.981 PORTD_AddressBase [6/6]

```
#define PORTD_AddressBase 0x500F
```

Definition at line 56 of file STM8L101F3.h.

5.2.2.982 RST_AddressBase [1/6]

```
#define RST_AddressBase 0x50B0
```

Definition at line 59 of file STM8L101F2.h.

5.2.2.983 RST_AddressBase [2/6]

```
#define RST_AddressBase 0x50B0
```

Definition at line 59 of file STM8L101K3.h.

5.2.2.984 RST_AddressBase [3/6]

```
#define RST_AddressBase 0x50B0
```

Definition at line 59 of file STM8L101F1.h.

5.2.2.985 RST_AddressBase [4/6]

```
#define RST_AddressBase 0x50B0
```

Definition at line 59 of file STM8L101G2.h.

5.2.2.986 RST_AddressBase [5/6]

```
#define RST_AddressBase 0x50B0
```

Definition at line 59 of file STM8L101G3.h.

5.2.2.987 RST_AddressBase [6/6]

```
#define RST_AddressBase 0x50B0
```

Definition at line 59 of file STM8L101F3.h.

5.2.2.988 SPI_AddressBase [1/6]

```
#define SPI_AddressBase 0x5200
```

Definition at line 64 of file STM8L101F2.h.

5.2.2.989 SPI_AddressBase [2/6]

```
#define SPI_AddressBase 0x5200
```

Definition at line 64 of file STM8L101F3.h.

5.2.2.990 SPI_AddressBase [3/6]

```
#define SPI_AddressBase 0x5200
```

Definition at line 64 of file STM8L101G2.h.

5.2.2.991 SPI_AddressBase [4/6]

```
#define SPI_AddressBase 0x5200
```

Definition at line 64 of file STM8L101F1.h.

5.2.2.992 SPI_AddressBase [5/6]

```
#define SPI_AddressBase 0x5200
```

Definition at line 64 of file STM8L101K3.h.

5.2.2.993 SPI_AddressBase [6/6]

```
#define SPI_AddressBase 0x5200
```

Definition at line 64 of file STM8L101G3.h.

5.2.2.994 STM8_ADDR_WIDTH

```
#define STM8_ADDR_WIDTH 16
```

width of address space

Definition at line 81 of file STM8L10x.h.

5.2.2.995 STM8_EEPROM_END

```
#define STM8_EEPROM_END (STM8_EEPROM_START + STM8_EEPROM_SIZE - 1)
```

last address in EEPROM

Definition at line 77 of file STM8L10x.h.

5.2.2.996 STM8_EEPROM_SIZE [1/7]

```
#define STM8_EEPROM_SIZE 0
```

Definition at line 49 of file STM8L101F3.h.

5.2.2.997 STM8_EEPROM_SIZE [2/7]

```
#define STM8_EEPROM_SIZE 0
```

Definition at line 49 of file STM8L101G3.h.

5.2.2.998 STM8_EEPROM_SIZE [3/7]

```
#define STM8_EEPROM_SIZE 0
```

Definition at line 49 of file STM8L101G2.h.

5.2.2.999 STM8_EEPROM_SIZE [4/7]

```
#define STM8_EEPROM_SIZE 0
```

Definition at line 49 of file STM8L101F1.h.

5.2.2.1000 STM8_EEPROM_SIZE [5/7]

```
#define STM8_EEPROM_SIZE 0
```

Definition at line 49 of file STM8L101K3.h.

5.2.2.1001 STM8_EEPROM_SIZE [6/7]

```
#define STM8_EEPROM_SIZE 0
```

Definition at line 49 of file STM8L101F2.h.

5.2.2.1002 STM8_EEPROM_SIZE [7/7]

```
#define STM8_EEPROM_SIZE 0
```

size of data EEPROM [B]

Definition at line 68 of file STM8L10x.h.

5.2.2.1003 STM8_EEPROM_START

```
#define STM8_EEPROM_START 9800
```

first address in EEPROM

Definition at line 76 of file STM8L10x.h.

5.2.2.1004 STM8_MEM_POINTER_T

```
#define STM8_MEM_POINTER_T uint16_t
```

address variable type

Definition at line 82 of file STM8L10x.h.

5.2.2.1005 STM8_PFLASH_END

```
#define STM8_PFLASH_END (STM8_PFLASH_START + STM8_PFLASH_SIZE - 1)
```

last address in program flash

Definition at line 73 of file STM8L10x.h.

5.2.2.1006 STM8_PFLASH_SIZE [1/7]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8L101F3.h.

5.2.2.1007 STM8_PFLASH_SIZE [2/7]

```
#define STM8_PFLASH_SIZE 4096
```

Definition at line 47 of file STM8L101F2.h.

5.2.2.1008 STM8_PFLASH_SIZE [3/7]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8L101K3.h.

5.2.2.1009 STM8_PFLASH_SIZE [4/7]

```
#define STM8_PFLASH_SIZE 2048
```

Definition at line 47 of file STM8L101F1.h.

5.2.2.1010 STM8_PFLASH_SIZE [5/7]

```
#define STM8_PFLASH_SIZE 4096
```

Definition at line 47 of file STM8L101G2.h.

5.2.2.1011 STM8_PFLASH_SIZE [6/7]

```
#define STM8_PFLASH_SIZE 8192
```

Definition at line 47 of file STM8L101G3.h.

5.2.2.1012 STM8_PFLASH_SIZE [7/7]

```
#define STM8_PFLASH_SIZE 2048
```

size of program flash [B]

Definition at line 60 of file STM8L10x.h.

5.2.2.1013 STM8_PFLASH_START

```
#define STM8_PFLASH_START 0x8000
```

first address in program flash

Definition at line 72 of file STM8L10x.h.

5.2.2.1014 STM8_RAM_END

```
#define STM8_RAM_END (STM8_RAM_START + STM8_RAM_SIZE - 1)
```

last address in RAM

Definition at line 75 of file STM8L10x.h.

5.2.2.1015 STM8_RAM_SIZE [1/7]

```
#define STM8_RAM_SIZE 1536
```

Definition at line 48 of file STM8L101K3.h.

5.2.2.1016 STM8_RAM_SIZE [2/7]

```
#define STM8_RAM_SIZE 1536
```

Definition at line 48 of file STM8L101F1.h.

5.2.2.1017 STM8_RAM_SIZE [3/7]

```
#define STM8_RAM_SIZE 1536
```

Definition at line 48 of file STM8L101G2.h.

5.2.2.1018 STM8_RAM_SIZE [4/7]

```
#define STM8_RAM_SIZE 1536
```

Definition at line 48 of file STM8L101F3.h.

5.2.2.1019 STM8_RAM_SIZE [5/7]

```
#define STM8_RAM_SIZE 1536
```

Definition at line 48 of file STM8L101G3.h.

5.2.2.1020 STM8_RAM_SIZE [6/7]

```
#define STM8_RAM_SIZE 1536
```

Definition at line 48 of file STM8L101F2.h.

5.2.2.1021 STM8_RAM_SIZE [7/7]

```
#define STM8_RAM_SIZE 1536
```

size of RAM [B]

Definition at line 64 of file STM8L10x.h.

5.2.2.1022 STM8_RAM_START

```
#define STM8_RAM_START 0x0000
```

first address in RAM

Definition at line 74 of file STM8L10x.h.

5.2.2.1023 STM8L101F1

```
#define STM8L101F1
```

Definition at line 40 of file STM8L101F1.h.

5.2.2.1024 STM8L101F2

```
#define STM8L101F2
```

Definition at line 40 of file STM8L101F2.h.

5.2.2.1025 STM8L101F3

```
#define STM8L101F3
```

Definition at line 40 of file STM8L101F3.h.

5.2.2.1026 STM8L101G2

```
#define STM8L101G2
```

Definition at line 40 of file STM8L101G2.h.

5.2.2.1027 STM8L101G3

```
#define STM8L101G3
```

Definition at line 40 of file STM8L101G3.h.

5.2.2.1028 STM8L101K3

```
#define STM8L101K3
```

Definition at line 40 of file STM8L101K3.h.

5.2.2.1029 STM8L10x [1/6]

```
#define STM8L10x
```

Definition at line 43 of file STM8L101F3.h.

5.2.2.1030 STM8L10x [2/6]

```
#define STM8L10x
```

Definition at line 43 of file STM8L101F2.h.

5.2.2.1031 STM8L10x [3/6]

```
#define STM8L10x
```

Definition at line 43 of file STM8L101F1.h.

5.2.2.1032 STM8L10x [4/6]

```
#define STM8L10x
```

Definition at line 43 of file STM8L101G3.h.

5.2.2.1033 STM8L10x [5/6]

```
#define STM8L10x
```

Definition at line 43 of file STM8L101K3.h.

5.2.2.1034 STM8L10x [6/6]

```
#define STM8L10x
```

Definition at line 43 of file STM8L101G2.h.

5.2.2.1035 SW_RESET

```
#define SW_RESET( ) (_WWDG_CR=0xBF)
```

reset controller via WWGD module

Definition at line 174 of file STM8L10x.h.

5.2.2.1036 TIM2_AddressBase [1/6]

```
#define TIM2_AddressBase 0x5250
```

Definition at line 68 of file STM8L101F3.h.

5.2.2.1037 TIM2_AddressBase [2/6]

```
#define TIM2_AddressBase 0x5250
```

Definition at line 68 of file STM8L101G3.h.

5.2.2.1038 TIM2_AddressBase [3/6]

```
#define TIM2_AddressBase 0x5250
```

Definition at line 68 of file STM8L101K3.h.

5.2.2.1039 TIM2_AddressBase [4/6]

```
#define TIM2_AddressBase 0x5250
```

Definition at line 68 of file STM8L101F2.h.

5.2.2.1040 TIM2_AddressBase [5/6]

```
#define TIM2_AddressBase 0x5250
```

Definition at line 68 of file STM8L101G2.h.

5.2.2.1041 TIM2_AddressBase [6/6]

```
#define TIM2_AddressBase 0x5250
```

Definition at line 68 of file STM8L101F1.h.

5.2.2.1042 TIM3_AddressBase [1/6]

```
#define TIM3_AddressBase 0x5280
```

Definition at line 69 of file STM8L101F1.h.

5.2.2.1043 TIM3_AddressBase [2/6]

```
#define TIM3_AddressBase 0x5280
```

Definition at line 69 of file STM8L101F3.h.

5.2.2.1044 TIM3_AddressBase [3/6]

```
#define TIM3_AddressBase 0x5280
```

Definition at line 69 of file STM8L101G3.h.

5.2.2.1045 TIM3_AddressBase [4/6]

```
#define TIM3_AddressBase 0x5280
```

Definition at line 69 of file STM8L101F2.h.

5.2.2.1046 TIM3_AddressBase [5/6]

```
#define TIM3_AddressBase 0x5280
```

Definition at line 69 of file STM8L101K3.h.

5.2.2.1047 TIM3_AddressBase [6/6]

```
#define TIM3_AddressBase 0x5280
```

Definition at line 69 of file STM8L101G2.h.

5.2.2.1048 TIM4_AddressBase [1/6]

```
#define TIM4_AddressBase 0x52E0
```

Definition at line 70 of file STM8L101G3.h.

5.2.2.1049 TIM4_AddressBase [2/6]

```
#define TIM4_AddressBase 0x52E0
```

Definition at line 70 of file STM8L101F1.h.

5.2.2.1050 TIM4_AddressBase [3/6]

```
#define TIM4_AddressBase 0x52E0
```

Definition at line 70 of file STM8L101F3.h.

5.2.2.1051 TIM4_AddressBase [4/6]

```
#define TIM4_AddressBase 0x52E0
```

Definition at line 70 of file STM8L101F2.h.

5.2.2.1052 TIM4_AddressBase [5/6]

```
#define TIM4_AddressBase 0x52E0
```

Definition at line 70 of file STM8L101K3.h.

5.2.2.1053 TIM4_AddressBase [6/6]

```
#define TIM4_AddressBase 0x52E0
```

Definition at line 70 of file STM8L101G2.h.

5.2.2.1054 TRIGGER_TRAP

```
#define TRIGGER_TRAP __asm__("trap")
```

trigger a trap (=soft interrupt) e.g. for EMC robustness (see AN1015)

Definition at line 171 of file STM8L10x.h.

5.2.2.1055 UID_AddressBase [1/6]

```
#define UID_AddressBase 0x4925
```

Definition at line 76 of file STM8L101F2.h.

5.2.2.1056 UID_AddressBase [2/6]

```
#define UID_AddressBase 0x4925
```

Definition at line 76 of file STM8L101K3.h.

5.2.2.1057 UID_AddressBase [3/6]

```
#define UID_AddressBase 0x4925
```

Definition at line 76 of file STM8L101G3.h.

5.2.2.1058 UID_AddressBase [4/6]

```
#define UID_AddressBase 0x4925
```

Definition at line 76 of file STM8L101F1.h.

5.2.2.1059 UID_AddressBase [5/6]

```
#define UID_AddressBase 0x4925
```

Definition at line 76 of file STM8L101G2.h.

5.2.2.1060 UID_AddressBase [6/6]

```
#define UID_AddressBase 0x4925
```

Definition at line 76 of file STM8L101F3.h.

5.2.2.1061 USART_AddressBase [1/6]

```
#define USART_AddressBase 0x5230
```

Definition at line 66 of file STM8L101F3.h.

5.2.2.1062 USART_AddressBase [2/6]

```
#define USART_AddressBase 0x5230
```

Definition at line 66 of file STM8L101F2.h.

5.2.2.1063 USART_AddressBase [3/6]

```
#define USART_AddressBase 0x5230
```

Definition at line 66 of file STM8L101G2.h.

5.2.2.1064 USART_AddressBase [4/6]

```
#define USART_AddressBase 0x5230
```

Definition at line 66 of file STM8L101F1.h.

5.2.2.1065 USART_AddressBase [5/6]

```
#define USART_AddressBase 0x5230
```

Definition at line 66 of file STM8L101K3.h.

5.2.2.1066 USART_AddressBase [6/6]

```
#define USART_AddressBase 0x5230
```

Definition at line 66 of file STM8L101G3.h.

5.2.2.1067 WAIT_FOR_INTERRUPT

```
#define WAIT_FOR_INTERRUPT( ) __asm__("wfi")
```

stop code execution and wait for interrupt

Definition at line 172 of file STM8L10x.h.

5.2.2.1068 WFE_AddressBase [1/6]

```
#define WFE_AddressBase 0x50A6
```

Definition at line 67 of file STM8L101G2.h.

5.2.2.1069 WFE_AddressBase [2/6]

```
#define WFE_AddressBase 0x50A6
```

Definition at line 67 of file STM8L101K3.h.

5.2.2.1070 WFE_AddressBase [3/6]

```
#define WFE_AddressBase 0x50A6
```

Definition at line 67 of file STM8L101G3.h.

5.2.2.1071 WFE_AddressBase [4/6]

```
#define WFE_AddressBase 0x50A6
```

Definition at line 67 of file STM8L101F1.h.

5.2.2.1072 WFE_AddressBase [5/6]

```
#define WFE_AddressBase 0x50A6
```

Definition at line 67 of file STM8L101F2.h.

5.2.2.1073 WFE_AddressBase [6/6]

```
#define WFE_AddressBase 0x50A6
```

Definition at line 67 of file STM8L101F3.h.

Chapter 6

Data Structure Documentation

6.1 ADC1_t Struct Reference

struct containing Analog Digital Converter 1 (ADC1)

`#include <STM8AF_STM8S.h>`

Data Fields

- struct {
  _BITS DATA: 2
  Data buffer 0 value [9:8].
  _BITS __pad0__: 6
  } DB0RH

  ADC1 10-bit Data Buffer Register 0 (ADC1_DB0RH)
- struct {
  _BITS DATA: 8
  Data buffer 0 value (low)
  } DB0RL

  ADC1 10-bit Data Buffer Register 0 (ADC1_DB0RL)
- struct {
  _BITS DATA: 2
  Data buffer 1 value [9:8].
  _BITS __pad0__: 6
  } DB1RH

  ADC1 10-bit Data Buffer Register 1 (ADC1_DB1RH)
- struct {
  _BITS DATA: 8
  Data buffer 1 value (low)
  } DB1RL

  ADC1 10-bit Data Buffer Register 1 (ADC1_DB1RL)

- struct {
  _BITS DATA: 2
  Data buffer 2 value [9:8].
  _BITS __pad0__: 6
  } DB2RH

ADC1 10-bit Data Buffer Register 2 (ADC1_DB2RH)

- struct {
  _BITS DATA: 8
  Data buffer 2 value (low)
  } DB2RL

ADC1 10-bit Data Buffer Register 2 (ADC1_DB2RL)

- struct {
  _BITS DATA: 2
  Data buffer 3 value [9:8].
  _BITS __pad0__: 6
  } DB3RH

ADC1 10-bit Data Buffer Register 3 (ADC1_DB3RH)

- struct {
  _BITS DATA: 8
  Data buffer 3 value (low)
  } DB3RL

ADC1 10-bit Data Buffer Register 3 (ADC1_DB3RL)

- struct {
  _BITS DATA: 2
  Data buffer 4 value [9:8].
  _BITS __pad0__: 6
  } DB4RH

ADC1 10-bit Data Buffer Register 4 (ADC1_DB4RH)

- struct {
  _BITS DATA: 8
  Data buffer 4 value (low)
  } DB4RL

ADC1 10-bit Data Buffer Register 4 (ADC1_DB4RL)

- struct {
  _BITS DATA: 2
  Data buffer 5 value [9:8].
  _BITS __pad0__: 6
  } DB5RH

ADC1 10-bit Data Buffer Register 5 (ADC1_DB5RH)

- struct {
  _BITS DATA: 8
  Data buffer 5 value (low)
  } DB5RL

ADC1 10-bit Data Buffer Register 5 (ADC1_DB5RL)

- struct {
  _BITS DATA: 2
  Data buffer 6 value [9:8].
  _BITS __pad0__: 6

} DB6RH

ADC1 10-bit Data Buffer Register 6 (ADC1_DB6RH)

- struct {
  _BITS DATA: 8
  Data buffer 6 value (low)
  } DB6RL

ADC1 10-bit Data Buffer Register 6 (ADC1_DB6RL)

- struct {
  _BITS DATA: 2
  Data buffer 7 value [9:8].
  _BITS __pad0__: 6
  } DB7RH

ADC1 10-bit Data Buffer Register 7 (ADC1_DB7RH)

- struct {
  _BITS DATA: 8
  Data buffer 7 value (low)
  } DB7RL

ADC1 10-bit Data Buffer Register 7 (ADC1_DB7RL)

- struct {
  _BITS DATA: 2
  Data buffer 8 value [9:8].
  _BITS __pad0__: 6
  } DB8RH

ADC1 10-bit Data Buffer Register 8 (ADC1_DB8RH)

- struct {
  _BITS DATA: 8
  Data buffer 8 value (low)
  } DB8RL

ADC1 10-bit Data Buffer Register 8 (ADC1_DB8RL)

- struct {
  _BITS DATA: 2
  Data buffer 9 value [9:8].
  _BITS __pad0__: 6
  } DB9RH

ADC1 10-bit Data Buffer Register 9 (ADC1_DB9RH)

- struct {
  _BITS DATA: 8
  Data buffer 9 value (low)
  } DB9RL

ADC1 10-bit Data Buffer Register 9 (ADC1_DB8RL)

- uint8_t res [12]

Reserved register (12B)

- struct {
  _BITS CH: 4
  Channel selection bits.
  _BITS AWDIE: 1
  Analog watchdog interrupt enable.
  _BITS EOCIE: 1

Interrupt enable for EOC.
_BITS AWD: 1
Analog Watchdog flag.
_BITS EOC: 1
End of conversion.
} CSR

ADC1 control/status register (ADC1_CSR)

- struct {
_BITS ADON: 1
A/D Converter on/off.
_BITS CONT: 1
Continuous conversion.
_BITS __pad0__: 2
_BITS SPSEL: 3
Clock prescaler selection.
_BITS __pad1__: 1
} CR1

ADC1 Configuration Register 1 (ADC1_CR1)

- struct {
_BITS __pad0__: 1
_BITS SCAN: 1
Scan mode enable.
_BITS __pad1__: 1
_BITS ALIGN: 1
Data alignment.
_BITS EXTSEL: 2
External event selection.
_BITS EXTTRIG: 1
External trigger enable.
_BITS __pad2__: 1
} CR2

ADC1 Configuration Register 2 (ADC1_CR2)

- struct {
_BITS __pad0__: 6
_BITS OVR: 1
Overrun flag.
_BITS DBUF: 1
Data buffer enable.
} CR3

ADC1 Configuration Register 3 (ADC1_CR3)

- struct {
_BITS DATA: 2
Data value [9:8].
_BITS __pad0__: 6
} DRH

ADC1 (unbuffered) 10-bit measurement result (ADC1_DRH)

- struct {
_BITS DATA: 8
Data value [7:0].
} DRL

ADC1 (unbuffered) 10-bit measurement result (ADC1_DRL)

- struct {
  _BITS TDH: 8
  Schmitt trigger disable [15:8].
  } TDRH

ADC1 Schmitt trigger disable register (ADC1_TDRH)

- struct {
  _BITS TDL: 8
  Schmitt trigger disable [7:0].
  } TDRL

ADC1 Schmitt trigger disable register (ADC1_TDRL)

- struct {
  _BITS HT: 8
  watchdog high threshold [9:2]
  } HTRH

ADC1 watchdog high threshold register (ADC1_HTRH)

- struct {
  _BITS HT: 2
  watchdog high threshold [1:0]
  _BITS __pad0__: 6
  } HTRL

ADC1 watchdog high threshold register (ADC1_HTRL)

- struct {
  _BITS LT: 8
  watchdog low threshold [9:2]
  } LTRH

ADC1 watchdog low threshold register (ADC1_LTRH)

- struct {
  _BITS LT: 2
  watchdog low threshold [1:0]
  _BITS __pad0__: 6
  } LTRL

ADC1 watchdog low threshold register (ADC1_LTRL)

- struct {
  _BITS AWS: 2
  watchdog status register [9:8]
  _BITS __pad0__: 6
  } AWSRH

ADC1 watchdog status register (ADC1_AWSRH)

- struct {
  _BITS AWS: 8
  watchdog status register [7:0]
  } AWSRL

ADC1 watchdog status register (ADC1_AWSRL)

- struct {
  _BITS AWEN: 2
  watchdog control register [9:8]
  _BITS __pad0__: 6
  } AWCRH

ADC1 watchdog control register (ADC1_AWCRH)

- struct {
  _BITS AWEN: 8
  watchdog control register [7:0]
  } AWCRL

ADC1 watchdog control register (ADC1_AWCRL)

6.1.1 Detailed Description

struct containing Analog Digital Converter 1 (ADC1)

Definition at line 4539 of file STM8AF_STM8S.h.

6.1.2 Field Documentation

6.1.2.1 __pad0__

_BITS __pad0__

Definition at line 4544 of file STM8AF_STM8S.h.

6.1.2.2 __pad1__

_BITS __pad1__

Definition at line 4690 of file STM8AF_STM8S.h.

6.1.2.3 __pad2__

_BITS __pad2__

Definition at line 4702 of file STM8AF_STM8S.h.

6.1.2.4 ADON

_BITS ADON

A/D Converter on/off.

Definition at line 4686 of file STM8AF_STM8S.h.

6.1.2.5 ALIGN

```
_BITS ALIGN
```

Data alignment.

Definition at line 4699 of file STM8AF_STM8S.h.

6.1.2.6 AWCRH

```
struct { ...  } AWCRH
```

ADC1 watchdog control register (ADC1_AWCRH)

6.1.2.7 AWCRL

```
struct { ...  } AWCRL
```

ADC1 watchdog control register (ADC1_AWCRL)

6.1.2.8 AWD

```
_BITS AWD
```

Analog Watchdog flag.

Definition at line 4679 of file STM8AF_STM8S.h.

6.1.2.9 AWDIE

```
_BITS AWDIE
```

Analog watchdog interrupt enable.

Definition at line 4677 of file STM8AF_STM8S.h.

6.1.2.10 AWEN

_BITS AWEN

watchdog control register [9:8]

watchdog control register [7:0]

Definition at line 4781 of file STM8AF_STM8S.h.

6.1.2.11 AWS

_BITS AWS

watchdog status register [9:8]

watchdog status register [7:0]

Definition at line 4768 of file STM8AF_STM8S.h.

6.1.2.12 AWSRH

struct { ... } AWSRH

ADC1 watchdog status register (ADC1_AWSRH)

6.1.2.13 AWSRL

struct { ... } AWSRL

ADC1 watchdog status register (ADC1_AWSRL)

6.1.2.14 CH

_BITS CH

Channel selection bits.

Definition at line 4676 of file STM8AF_STM8S.h.

6.1.2.15 CONT

```
_BITS CONT
```

Continuous conversion.

Definition at line 4687 of file STM8AF_STM8S.h.

6.1.2.16 CR1

```
struct { ...  } CR1
```

ADC1 Configuration Register 1 (ADC1_CR1)

6.1.2.17 CR2

```
struct { ...  } CR2
```

ADC1 Configuration Register 2 (ADC1_CR2)

6.1.2.18 CR3

```
struct { ...  } CR3
```

ADC1 Configuration Register 3 (ADC1_CR3)

6.1.2.19 CSR

```
struct { ...  } CSR
```

ADC1 control/status register (ADC1_CSR)

6.1.2.20 DATA

_BITS DATA

Data buffer 0 value [9:8].

Data value [7:0].

Data value [9:8].

Data buffer 9 value (low)

Data buffer 9 value [9:8].

Data buffer 8 value (low)

Data buffer 8 value [9:8].

Data buffer 7 value (low)

Data buffer 7 value [9:8].

Data buffer 6 value (low)

Data buffer 6 value [9:8].

Data buffer 5 value (low)

Data buffer 5 value [9:8].

Data buffer 4 value (low)

Data buffer 4 value [9:8].

Data buffer 3 value (low)

Data buffer 3 value [9:8].

Data buffer 2 value (low)

Data buffer 2 value [9:8].

Data buffer 1 value (low)

Data buffer 1 value [9:8].

Data buffer 0 value (low)

Definition at line 4543 of file STM8AF_STM8S.h.

6.1.2.21 DB0RH

struct { ... } DB0RH

ADC1 10-bit Data Buffer Register 0 (ADC1_DB0RH)

6.1.2.22 DB0RL

```
struct { ...  } DB0RL
```

ADC1 10-bit Data Buffer Register 0 (ADC1_DB0RL)

6.1.2.23 DB1RH

```
struct { ...  } DB1RH
```

ADC1 10-bit Data Buffer Register 1 (ADC1_DB1RH)

6.1.2.24 DB1RL

```
struct { ...  } DB1RL
```

ADC1 10-bit Data Buffer Register 1 (ADC1_DB1RL)

6.1.2.25 DB2RH

```
struct { ...  } DB2RH
```

ADC1 10-bit Data Buffer Register 2 (ADC1_DB2RH)

6.1.2.26 DB2RL

```
struct { ...  } DB2RL
```

ADC1 10-bit Data Buffer Register 2 (ADC1_DB2RL)

6.1.2.27 DB3RH

```
struct { ...  } DB3RH
```

ADC1 10-bit Data Buffer Register 3 (ADC1_DB3RH)

6.1.2.28 DB3RL

```
struct { ... } DB3RL
```

ADC1 10-bit Data Buffer Register 3 (ADC1_DB3RL)

6.1.2.29 DB4RH

```
struct { ... } DB4RH
```

ADC1 10-bit Data Buffer Register 4 (ADC1_DB4RH)

6.1.2.30 DB4RL

```
struct { ... } DB4RL
```

ADC1 10-bit Data Buffer Register 4 (ADC1_DB4RL)

6.1.2.31 DB5RH

```
struct { ... } DB5RH
```

ADC1 10-bit Data Buffer Register 5 (ADC1_DB5RH)

6.1.2.32 DB5RL

```
struct { ... } DB5RL
```

ADC1 10-bit Data Buffer Register 5 (ADC1_DB5RL)

6.1.2.33 DB6RH

```
struct { ... } DB6RH
```

ADC1 10-bit Data Buffer Register 6 (ADC1_DB6RH)

6.1.2.34 DB6RL

```
struct { ... } DB6RL
```

ADC1 10-bit Data Buffer Register 6 (ADC1_DB6RL)

6.1.2.35 DB7RH

```
struct { ... } DB7RH
```

ADC1 10-bit Data Buffer Register 7 (ADC1_DB7RH)

6.1.2.36 DB7RL

```
struct { ... } DB7RL
```

ADC1 10-bit Data Buffer Register 7 (ADC1_DB7RL)

6.1.2.37 DB8RH

```
struct { ... } DB8RH
```

ADC1 10-bit Data Buffer Register 8 (ADC1_DB8RH)

6.1.2.38 DB8RL

```
struct { ... } DB8RL
```

ADC1 10-bit Data Buffer Register 8 (ADC1_DB8RL)

6.1.2.39 DB9RH

```
struct { ... } DB9RH
```

ADC1 10-bit Data Buffer Register 9 (ADC1_DB9RH)

6.1.2.40 DB9RL

```
struct { ...  } DB9RL
```

ADC1 10-bit Data Buffer Register 9 (ADC1_DB8RL)

6.1.2.41 DBUF

```
_BITS DBUF
```

Data buffer enable.

Definition at line 4710 of file STM8AF_STM8S.h.

6.1.2.42 DRH

```
struct { ...  } DRH
```

ADC1 (unbuffered) 10-bit measurement result (ADC1_DRH)

6.1.2.43 DRL

```
struct { ...  } DRL
```

ADC1 (unbuffered) 10-bit measurement result (ADC1_DRL)

6.1.2.44 EOC

```
_BITS EOC
```

End of conversion.

Definition at line 4680 of file STM8AF_STM8S.h.

6.1.2.45 EOCIE

```
_BITS EOCIE
```

Interrupt enable for EOC.

Definition at line 4678 of file STM8AF_STM8S.h.

6.1.2.46 EXTSEL

```
_BITS EXTSEL
```

External event selection.

Definition at line 4700 of file STM8AF_STM8S.h.

6.1.2.47 EXTTRIG

```
_BITS EXTTRIG
```

External trigger enable.

Definition at line 4701 of file STM8AF_STM8S.h.

6.1.2.48 HT

```
_BITS HT
```

watchdog high threshold [9:2]

watchdog high threshold [1:0]

Definition at line 4741 of file STM8AF_STM8S.h.

6.1.2.49 HTRH

```
struct { ...  } HTRH
```

ADC1 watchdog high threshold register (ADC1_HTRH)

6.1.2.50 HTRL

```
struct { ...  } HTRL
```

ADC1 watchdog high threshold register (ADC1_HTRL)

6.1.2.51 LT

_BITS LT

watchdog low threshold [9:2]

watchdog low threshold [1:0]

Definition at line 4755 of file STM8AF_STM8S.h.

6.1.2.52 LTRH

struct { ... } LTRH

ADC1 watchdog low threshold register (ADC1_LTRH)

6.1.2.53 LTRL

struct { ... } LTRL

ADC1 watchdog low threshold register (ADC1_LTRL)

6.1.2.54 OVR

_BITS OVR

Overrun flag.

Definition at line 4709 of file STM8AF_STM8S.h.

6.1.2.55 res

uint8_t res[12]

Reserved register (12B)

Definition at line 4671 of file STM8AF_STM8S.h.

6.1.2.56 SCAN

_BITS SCAN

Scan mode enable.

Definition at line 4697 of file STM8AF_STM8S.h.

6.1.2.57 SPSEL

_BITS SPSEL

Clock prescaler selection.

Definition at line 4689 of file STM8AF_STM8S.h.

6.1.2.58 TDH

_BITS TDH

Schmitt trigger disable [15:8].

Definition at line 4729 of file STM8AF_STM8S.h.

6.1.2.59 TDL

_BITS TDL

Schmitt trigger disable [7:0].

Definition at line 4735 of file STM8AF_STM8S.h.

6.1.2.60 TDRH

struct { ... } TDRH

ADC1 Schmitt trigger disable register (ADC1_TDRH)

6.1.2.61 TDRL

```
struct { ...  } TDRL
```

ADC1 Schmitt trigger disable register (ADC1_TDRL)

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

6.2 ADC2_t Struct Reference

struct containing Analog Digital Converter 2 (ADC2)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS CH: 4
  Channel selection bits.
  _BITS __pad0__: 1
  _BITS EOCIE: 1
  Interrupt enable for EOC.
  _BITS __pad1__: 1
  _BITS EOC: 1
  End of conversion.
  } CSR

  ADC2 control/status register (ADC2_CSR)
- struct {
  _BITS ADON: 1
  A/D Converter on/off.
  _BITS CONT: 1
  Continuous conversion.
  _BITS __pad0__: 2
  _BITS SPSEL: 3
  Clock prescaler selection.
  _BITS __pad1__: 1
  } CR1

  ADC2 Configuration Register 1 (ADC2_CR1)
- struct {
  _BITS __pad0__: 3
  _BITS ALIGN: 1
  Data alignment.
  _BITS EXTSEL: 2
  External event selection.
  _BITS EXTTRIG: 1
  External trigger enable.
  _BITS __pad1__: 1
  } CR2

ADC2 Configuration Register 2 (ADC2_CR2)

- uint8_t res [1]

  Reserved register (1B)

- struct {
  _BITS DATA: 2
  Data value [9:8].
  _BITS __pad0__: 6
  } DRH

  ADC2 (unbuffered) 10-bit measurement result (ADC2_DRH)

- struct {
  _BITS DATA: 8
  Data value [7:0].
  } DRL

  ADC2 (unbuffered) 10-bit measurement result (ADC2_DRL)

- struct {
  _BITS TDH: 8
  Schmitt trigger disable [9:8].
  } TDRH

  ADC2 Schmitt trigger disable register (ADC2_TDRH)

- struct {
  _BITS TDL: 8
  Schmitt trigger disable [7:0].
  } TDRL

  ADC2 Schmitt trigger disable register (ADC2_TDRL)

6.2.1 Detailed Description

struct containing Analog Digital Converter 2 (ADC2)

Definition at line 4894 of file STM8AF_STM8S.h.

6.2.2 Field Documentation

6.2.2.1 __pad0__

```
_BITS __pad0__
```

Definition at line 4899 of file STM8AF_STM8S.h.

6.2.2.2 __pad1__

_BITS __pad1__

Definition at line 4901 of file STM8AF_STM8S.h.

6.2.2.3 ADON

_BITS ADON

A/D Converter on/off.

Definition at line 4908 of file STM8AF_STM8S.h.

6.2.2.4 ALIGN

_BITS ALIGN

Data alignment.

Definition at line 4919 of file STM8AF_STM8S.h.

6.2.2.5 CH

_BITS CH

Channel selection bits.

Definition at line 4898 of file STM8AF_STM8S.h.

6.2.2.6 CONT

_BITS CONT

Continuous conversion.

Definition at line 4909 of file STM8AF_STM8S.h.

6.2.2.7 CR1

```
struct { ...  } CR1
```

ADC2 Configuration Register 1 (ADC2_CR1)

6.2.2.8 CR2

```
struct { ...  } CR2
```

ADC2 Configuration Register 2 (ADC2_CR2)

6.2.2.9 CSR

```
struct { ...  } CSR
```

ADC2 control/status register (ADC2_CSR)

6.2.2.10 DATA

```
_BITS DATA
```

Data value [9:8].

Data value [7:0].

Definition at line 4932 of file STM8AF_STM8S.h.

6.2.2.11 DRH

```
struct { ...  } DRH
```

ADC2 (unbuffered) 10-bit measurement result (ADC2_DRH)

6.2.2.12 DRL

```
struct { ...  } DRL
```

ADC2 (unbuffered) 10-bit measurement result (ADC2_DRL)

6.2.2.13 EOC

_BITS EOC

End of conversion.

Definition at line 4902 of file STM8AF_STM8S.h.

6.2.2.14 EOCIE

_BITS EOCIE

Interrupt enable for EOC.

Definition at line 4900 of file STM8AF_STM8S.h.

6.2.2.15 EXTSEL

_BITS EXTSEL

External event selection.

Definition at line 4920 of file STM8AF_STM8S.h.

6.2.2.16 EXTTRIG

_BITS EXTTRIG

External trigger enable.

Definition at line 4921 of file STM8AF_STM8S.h.

6.2.2.17 res

uint8_t res[1]

Reserved register (1B)

Definition at line 4927 of file STM8AF_STM8S.h.

6.2.2.18 SPSEL

_BITS SPSEL

Clock prescaler selection.

Definition at line 4911 of file STM8AF_STM8S.h.

6.2.2.19 TDH

_BITS TDH

Schmitt trigger disable [9:8].

Definition at line 4945 of file STM8AF_STM8S.h.

6.2.2.20 TDL

_BITS TDL

Schmitt trigger disable [7:0].

Definition at line 4951 of file STM8AF_STM8S.h.

6.2.2.21 TDRH

struct { ... } TDRH

ADC2 Schmitt trigger disable register (ADC2_TDRH)

6.2.2.22 TDRL

struct { ... } TDRL

ADC2 Schmitt trigger disable register (ADC2_TDRL)

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

6.3 AWU_t Struct Reference

struct for cofiguring the Auto Wake-Up Module (AWU)

`#include <STM8AF_STM8S.h>`

Data Fields

- struct {
  _BITS MSR: 1
  LSI measurement enable.
  _BITS __pad0__: 3
  _BITS AWUEN: 1
  Auto-wakeup enable.
  _BITS AWUF: 1
  Auto-wakeup flag.
  _BITS __pad1__: 2
  } CSR

  AWU Control/status register (AWU_CSR)
- struct {
  _BITS APRE: 6
  Asynchronous prescaler divider.
  _BITS __pad0__: 2
  } APR

  AWU Asynchronous prescaler register (AWU_APR)
- struct {
  _BITS AWUTB: 4
  Auto-wakeup timebase selection.
  _BITS __pad0__: 4
  } TBR

  AWU Timebase selection register (AWU_TBR)
- struct {
  _BITS MSR: 1
  LSI measurement enable.
  _BITS __pad0__: 3
  _BITS AWUEN: 1
  Auto-wakeup enable.
  _BITS AWUF: 1
  Auto-wakeup flag.
  _BITS __pad1__: 2
  } CSR

  AWU Control/status register (AWU_CSR)
- struct {
  _BITS APRE: 6
  Asynchronous prescaler divider.
  _BITS __pad0__: 2
  } APR

  AWU Asynchronous prescaler register (AWU_APR)

- struct {
  _BITS AWUTB: 4
  Auto-wakeup timebase selection.
  _BITS __pad0__: 4
  } TBR

  AWU Timebase selection register (AWU_TBR)

6.3.1 Detailed Description

struct for cofiguring the Auto Wake-Up Module (AWU)

Definition at line 1100 of file STM8AF_STM8S.h.

6.3.2 Field Documentation

6.3.2.1 __pad0__

`_BITS __pad0__`

Definition at line 1105 of file STM8AF_STM8S.h.

6.3.2.2 __pad1__

`_BITS __pad1__`

Definition at line 1108 of file STM8AF_STM8S.h.

6.3.2.3 APR [1/2]

`struct { ... } APR`

AWU Asynchronous prescaler register (AWU_APR)

6.3.2.4 APR [2/2]

`struct { ... } APR`

AWU Asynchronous prescaler register (AWU_APR)

6.3.2.5 APRE

_BITS APRE

Asynchronous prescaler divider.

Definition at line 1114 of file STM8AF_STM8S.h.

6.3.2.6 AWUEN

_BITS AWUEN

Auto-wakeup enable.

Definition at line 1106 of file STM8AF_STM8S.h.

6.3.2.7 AWUF

_BITS AWUF

Auto-wakeup flag.

Definition at line 1107 of file STM8AF_STM8S.h.

6.3.2.8 AWUTB

_BITS AWUTB

Auto-wakeup timebase selection.

Definition at line 1121 of file STM8AF_STM8S.h.

6.3.2.9 CSR [1/2]

struct { ... } CSR

AWU Control/status register (AWU_CSR)

6.3.2.10 CSR [2/2]

```
struct { ...  } CSR
```

AWU Control/status register (AWU_CSR)

6.3.2.11 MSR

```
_BITS MSR
```

LSI measurement enable.

Definition at line 1104 of file STM8AF_STM8S.h.

6.3.2.12 TBR [1/2]

```
struct { ...  } TBR
```

AWU Timebase selection register (AWU_TBR)

6.3.2.13 TBR [2/2]

```
struct { ...  } TBR
```

AWU Timebase selection register (AWU_TBR)

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.4 BEEP_t Struct Reference

struct for beeper control (BEEP)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS BEEPDIV: 5
  Beep clock prescaler divider.
  _BITS BEEPEN: 1
  Beep enable.
  _BITS BEEPSEL: 2
  Beeper frequency selectiouint8_tn.
  } CSR

  Beeper control/status register (BEEP_CSR)
- struct {
  _BITS BEEPDIV: 5
  Beep clock prescaler divider.
  _BITS BEEPEN: 1
  Beep enable.
  _BITS BEEPSEL: 2
  Beeper frequency selection.
  } CSR

  Beeper control/status register (BEEP_CSR)

6.4.1 Detailed Description

struct for beeper control (BEEP)

Definition at line 1173 of file STM8AF_STM8S.h.

6.4.2 Field Documentation

6.4.2.1 BEEPDIV

```
_BITS BEEPDIV
```

Beep clock prescaler divider.

Definition at line 1177 of file STM8AF_STM8S.h.

6.4.2.2 BEEPEN

```
_BITS BEEPEN
```

Beep enable.

Definition at line 1178 of file STM8AF_STM8S.h.

6.4.2.3 BEEPSEL

_BITS BEEPSEL

Beeper frequency selectiouint8_tn.

Beeper frequency selection.

Definition at line 1179 of file STM8AF_STM8S.h.

6.4.2.4 CSR [1/2]

struct { ... } CSR

Beeper control/status register (BEEP_CSR)

6.4.2.5 CSR [2/2]

struct { ... } CSR

Beeper control/status register (BEEP_CSR)

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.5 CAN_t Struct Reference

struct for controlling Controller Area Network Module (CAN)

#include <STM8AF_STM8S.h>

Data Fields

- struct {
  _BITS INRQ: 1
  Initialization Request.
  _BITS SLEEP: 1
  Sleep Mode Request.
  _BITS TXFP: 1
  Transmit FIFO Priority.
  _BITS RFLM: 1
  Receive FIFO Locked Mode.
  _BITS NART: 1
  No Automatic Retransmission.
  _BITS AWUM: 1
  Automatic Wakeup Mode.
  _BITS ABOM: 1
  Automatic Bus-Off Management.
  _BITS TTCM: 1
  Time Triggered Communication Mode.
  } MCR

  CAN master control register (CAN_MCR)
- struct {
  _BITS INAK: 1
  Initialization Acknowledge.
  _BITS SLAK: 1
  Sleep Acknowledge.
  _BITS ERRI: 1
  Error Interrupt.
  _BITS WKUI: 1
  Wakeup Interrupt.
  _BITS TX: 1
  Transmit.
  _BITS RX: 1
  Receive.
  _BITS __pad0__: 2
  } MSR

  CAN master status register (CAN_MSR)
- struct {
  _BITS RQCP0: 1
  Request Completed for Mailbox 0.
  _BITS RQCP1: 1
  Request Completed for Mailbox 1.
  _BITS RQCP2: 1
  Request Completed for Mailbox 2.
  _BITS __pad0__: 1
  _BITS TXOK0: 1
  Transmission ok for Mailbox 0.
  _BITS TXOK1: 1
  Transmission ok for Mailbox 1.
  _BITS TXOK2: 1
  Transmission ok for Mailbox 2.
  _BITS __pad1__: 1
  } TSR

  CAN transmit status register (CAN_TSR)

- struct {
  _BITS CODE: 2
  Mailbox Code.
  _BITS TME0: 1
  Transmit Mailbox 0 Empty.
  _BITS TME1: 1
  Transmit Mailbox 1 Empty.
  _BITS TME2: 1
  Transmit Mailbox 2 Empty.
  _BITS LOW0: 1
  Lowest Priority Flag for Mailbox 0.
  _BITS LOW1: 1
  Lowest Priority Flag for Mailbox 1.
  _BITS LOW2: 1
  Lowest Priority Flag for Mailbox 2.
  } TPR

CAN transmit priority register (CAN_TPR)

- struct {
  _BITS FMP: 2
  FIFO Message Pending.
  _BITS __pad0__: 1
  _BITS FULL: 1
  FIFO Full.
  _BITS FOVR: 1
  FIFO Overrun.
  _BITS RFOM: 1
  Release FIFO Output Mailbox.
  _BITS __pad1__: 2
  } RFR

CAN receive FIFO register (CAN_RFR)

- struct {
  _BITS TMEIE: 1
  Transmit Mailbox Empty Interrupt Enable.
  _BITS FMPIE: 1
  FIFO Message Pending Interrupt Enable.
  _BITS FFIE: 1
  FIFO Full Interrupt Enable.
  _BITS FOVIE: 1
  FIFO Overrun Interrupt Enable.
  _BITS __pad0__: 3
  _BITS WKUIE: 1
  Wakeup Interrupt Enable.
  } IER

CAN interrupt enable register (CAN_IER)

- struct {
  _BITS LBKM: 1
  Loop back mode.
  _BITS SILM: 1
  Silent mode.
  _BITS SAMP: 1
  Last sample point.
  _BITS RX: 1
  CAN Rx Signal.
  _BITS TXM2E: 1
  TX Mailbox 2 enable.

_BITS __pad0__: 3
} DGR

CAN diagnosis register (CAN_DGR)

- struct {
  _BITS PS: 3
  Page select.
  _BITS __pad0__: 5
  } PSR

CAN page selection register for paged registers (CAN_PSR)

- union {
  struct {
  struct {
  _BITS TXRQ: 1
  Transmission mailbox request.
  _BITS ABRQ: 1
  Abort request for mailbox.
  _BITS RQCP: 1
  Request completed.
  _BITS TXOK: 1
  Transmission OK.
  _BITS ALST: 1
  Arbitration lost.
  _BITS TERR: 1
  Transmission error.
  _BITS __pad0__: 2
  } MCSR
  CAN message control/status register (CAN_MCSR)
  struct {
  _BITS DLC: 4
  Data length code.
  _BITS __pad0__: 3
  _BITS TGT: 1
  Transmit global time.
  } MDLCR
  CAN mailbox data length control register (CAN_MDLCR)
  struct {
  _BITS ID: 5
  STID[10:6] or EXID[28:24].
  _BITS RTR: 1
  Remote transmission request.
  _BITS IDE: 1
  Extended identifier.
  _BITS __pad0__: 1
  } MIDR1
  CAN mailbox identifier register 1 (CAN_MIDR1)
  struct {
  _BITS EXID: 2
  EXID[17:16].
  _BITS ID: 6
  STID[5:0] or EXID[23:18].
  } MIDR2
  CAN mailbox identifier register 2 (CAN_MIDR2)
  struct {
  _BITS EXID: 8
  EXID[15:8].
  } MIDR3

CAN mailbox identifier register 3 (CAN_MIDR3)
struct {
_BITS EXID: 8
EXID[7:0].
} MIDR4
CAN mailbox identifier register 4 (CAN_MIDR4)
struct {
_BITS DATA: 8
data[7:0]
} MDAR1
CAN mailbox data register 1 (CAN_MDAR1)
struct {
_BITS DATA: 8
data[7:0]
} MDAR2
CAN mailbox data register 2 (CAN_MDAR2)
struct {
_BITS DATA: 8
data[7:0]
} MDAR3
CAN mailbox data register 3 (CAN_MDAR3)
struct {
_BITS DATA: 8
data[7:0]
} MDAR4
CAN mailbox data register 4 (CAN_MDAR4)
struct {
_BITS DATA: 8
data[7:0]
} MDAR5
CAN mailbox data register 5 (CAN_MDAR5)
struct {
_BITS DATA: 8
data[7:0]
} MDAR6
CAN mailbox data register 6 (CAN_MDAR6)
struct {
_BITS DATA: 8
data[7:0]
} MDAR7
CAN mailbox data register 7 (CAN_MDAR7)
struct {
_BITS DATA: 8
data[7:0]
} MDAR8
CAN mailbox data register 8 (CAN_MDAR8)
struct {
_BITS TIME: 8
Message time stamp [7:0].
} MTSRL
CAN mailbox time stamp register low byte (CAN_MTSRL)
struct {
_BITS TIME: 8
Message time stamp [15:8].
} MTSRH
CAN mailbox time stamp register high byte (CAN_MTSRH)
} PAGE_0
CAN page 0: Tx Mailbox 0 (CAN.PAGE_0)

struct {
  struct {
    _BITS TXRQ: 1
      Transmission mailbox request.
    _BITS ABRQ: 1
      Abort request for mailbox.
    _BITS RQCP: 1
      Request completed.
    _BITS TXOK: 1
      Transmission OK.
    _BITS ALST: 1
      Arbitration lost.
    _BITS TERR: 1
      Transmission error.
    _BITS __pad0__: 2
  } MCSR
      CAN message control/status register (CAN_MCSR)
  struct {
    _BITS DLC: 4
      Data length code.
    _BITS __pad0__: 3
    _BITS TGT: 1
      Transmit global time.
  } MDLCR
      CAN mailbox data length control register (CAN_MDLCR)
  struct {
    _BITS ID: 5
      STID[10:6] or EXID[28:24].
    _BITS RTR: 1
      Remote transmission request.
    _BITS IDE: 1
      Extended identifier.
    _BITS __pad0__: 1
  } MIDR1
      CAN mailbox identifier register 1 (CAN_MIDR1)
  struct {
    _BITS EXID: 2
      EXID[17:16].
    _BITS ID: 6
      STID[5:0] or EXID[23:18].
  } MIDR2
      CAN mailbox identifier register 2 (CAN_MIDR2)
  struct {
    _BITS EXID: 8
      EXID[15:8].
  } MIDR3
      CAN mailbox identifier register 3 (CAN_MIDR3)
  struct {
    _BITS EXID: 8
      EXID[7:0].
  } MIDR4
      CAN mailbox identifier register 4 (CAN_MIDR4)
  struct {
    _BITS DATA: 8
      data[7:0]
  } MDAR1
      CAN mailbox data register 1 (CAN_MDAR1)
  struct {

_BITS DATA: 8
data[7:0]
} MDAR2
CAN mailbox data register 2 (CAN_MDAR2)
struct {
_BITS DATA: 8
data[7:0]
} MDAR3
CAN mailbox data register 3 (CAN_MDAR3)
struct {
_BITS DATA: 8
data[7:0]
} MDAR4
CAN mailbox data register 4 (CAN_MDAR4)
struct {
_BITS DATA: 8
data[7:0]
} MDAR5
CAN mailbox data register 5 (CAN_MDAR5)
struct {
_BITS DATA: 8
data[7:0]
} MDAR6
CAN mailbox data register 6 (CAN_MDAR6)
struct {
_BITS DATA: 8
data[7:0]
} MDAR7
CAN mailbox data register 7 (CAN_MDAR7)
struct {
_BITS DATA: 8
data[7:0]
} MDAR8
CAN mailbox data register 8 (CAN_MDAR8)
struct {
_BITS TIME: 8
Message time stamp [7:0].
} MTSRL
CAN mailbox time stamp register low byte (CAN_MTSRL)
struct {
_BITS TIME: 8
Message time stamp [15:8].
} MTSRH
CAN mailbox time stamp register high byte (CAN_MTSRH)
} PAGE_1
CAN page 1: Tx Mailbox 1 (CAN.PAGE_1)
struct {
struct {
_BITS DATA: 8
CAN reception filter 0/1.
} F0R1
CAN reception filter 0/1 (CAN_F0R1)
struct {
_BITS DATA: 8
CAN reception filter 0/2.
} F0R2
CAN reception filter 0/2 (CAN_F0R2)
struct {

__BITS DATA: 8
CAN reception filter 0/3.
} F0R3
CAN reception filter 0/3 (CAN_F0R3)
struct {
__BITS DATA: 8
CAN reception filter 0/4.
} F0R4
CAN reception filter 0/4 (CAN_F0R4)
struct {
__BITS DATA: 8
CAN reception filter 0/5.
} F0R5
CAN reception filter 0/5 (CAN_F0R5)
struct {
__BITS DATA: 8
CAN reception filter 0/6.
} F0R6
CAN reception filter 0/6 (CAN_F0R6)
struct {
__BITS DATA: 8
CAN reception filter 0/7.
} F0R7
CAN reception filter 0/7 (CAN_F0R7)
struct {
__BITS DATA: 8
CAN reception filter 0/8.
} F0R8
CAN reception filter 0/8 (CAN_F0R8)
struct {
__BITS DATA: 8
CAN reception filter 1/1.
} F1R1
CAN reception filter 1/1 (CAN_F1R1)
struct {
__BITS DATA: 8
CAN reception filter 1/2.
} F1R2
CAN reception filter 1/2 (CAN_F1R2)
struct {
__BITS DATA: 8
CAN reception filter 1/3.
} F1R3
CAN reception filter 1/3 (CAN_F1R3)
struct {
__BITS DATA: 8
CAN reception filter 1/4.
} F1R4
CAN reception filter 1/4 (CAN_F1R4)
struct {
__BITS DATA: 8
CAN reception filter 1/5.
} F1R5
CAN reception filter 1/5 (CAN_F1R5)
struct {
__BITS DATA: 8
CAN reception filter 1/6.
} F1R6

CAN reception filter 1/6 (CAN_F1R6)
struct {
_BITS DATA: 8
CAN reception filter 1/7.
} F1R7
CAN reception filter 1/7 (CAN_F1R7)
struct {
_BITS DATA: 8
CAN reception filter 1/8.
} F1R8
CAN reception filter 1/8 (CAN_F1R8)
} PAGE_2
CAN page 2: Acceptance Filter 0:1 (CAN.PAGE_2)
struct {
struct {
_BITS DATA: 8
CAN reception filter 2/1.
} F2R1
CAN reception filter 2/1 (CAN_F2R1)
struct {
_BITS DATA: 8
CAN reception filter 2/2.
} F2R2
CAN reception filter 2/2 (CAN_F2R2)
struct {
_BITS DATA: 8
CAN reception filter 2/3.
} F2R3
CAN reception filter 2/3 (CAN_F2R3)
struct {
_BITS DATA: 8
CAN reception filter 2/4.
} F2R4
CAN reception filter 2/4 (CAN_F2R4)
struct {
_BITS DATA: 8
CAN reception filter 2/5.
} F2R5
CAN reception filter 2/5 (CAN_F2R5)
struct {
_BITS DATA: 8
CAN reception filter 2/6.
} F2R6
CAN reception filter 2/6 (CAN_F2R6)
struct {
_BITS DATA: 8
CAN reception filter 2/7.
} F2R7
CAN reception filter 2/7 (CAN_F2R7)
struct {
_BITS DATA: 8
CAN reception filter 2/8.
} F2R8
CAN reception filter 2/8 (CAN_F2R8)
struct {
_BITS DATA: 8
CAN reception filter 3/1.
} F3R1

CAN reception filter 3/1 (CAN_F3R1)
struct {
_BITS DATA: 8
CAN reception filter 3/2.
} F3R2
CAN reception filter 3/2 (CAN_F3R2)
struct {
_BITS DATA: 8
CAN reception filter 3/3.
} F3R3
CAN reception filter 3/3 (CAN_F3R3)
struct {
_BITS DATA: 8
CAN reception filter 3/4.
} F3R4
CAN reception filter 3/4 (CAN_F3R4)
struct {
_BITS DATA: 8
CAN reception filter 3/5.
} F3R5
CAN reception filter 3/5 (CAN_F3R5)
struct {
_BITS DATA: 8
CAN reception filter 3/6.
} F3R6
CAN reception filter 3/6 (CAN_F3R6)
struct {
_BITS DATA: 8
CAN reception filter 3/7.
} F3R7
CAN reception filter 3/7 (CAN_F3R7)
struct {
_BITS DATA: 8
CAN reception filter 3/8.
} F3R8
CAN reception filter 3/8 (CAN_F3R8)
} PAGE_3
CAN page 3: Acceptance Filter 2:3 (CAN.PAGE_3)
struct {
struct {
_BITS DATA: 8
CAN reception filter 4/1.
} F4R1
CAN reception filter 4/1 (CAN_F4R1)
struct {
_BITS DATA: 8
CAN reception filter 4/2.
} F4R2
CAN reception filter 4/2 (CAN_F4R2)
struct {
_BITS DATA: 8
CAN reception filter 4/3.
} F4R3
CAN reception filter 4/3 (CAN_F4R3)
struct {
_BITS DATA: 8
CAN reception filter 4/4.
} F4R4

CAN reception filter 4/4 (CAN_F4R4)
struct {
_BITS DATA: 8
CAN reception filter 4/5.
} F4R5
CAN reception filter 4/5 (CAN_F4R5)
struct {
_BITS DATA: 8
CAN reception filter 4/6.
} F4R6
CAN reception filter 4/6 (CAN_F4R6)
struct {
_BITS DATA: 8
CAN reception filter 4/7.
} F4R7
CAN reception filter 4/7 (CAN_F4R7)
struct {
_BITS DATA: 8
CAN reception filter 4/8.
} F4R8
CAN reception filter 4/8 (CAN_F4R8)
struct {
_BITS DATA: 8
CAN reception filter 5/1.
} F5R1
CAN reception filter 5/1 (CAN_F5R1)
struct {
_BITS DATA: 8
CAN reception filter 5/2.
} F5R2
CAN reception filter 5/2 (CAN_F5R2)
struct {
_BITS DATA: 8
CAN reception filter 5/3.
} F5R3
CAN reception filter 5/3 (CAN_F5R3)
struct {
_BITS DATA: 8
CAN reception filter 5/4.
} F5R4
CAN reception filter 5/4 (CAN_F5R4)
struct {
_BITS DATA: 8
CAN reception filter 5/5.
} F5R5
CAN reception filter 5/5 (CAN_F5R5)
struct {
_BITS DATA: 8
CAN reception filter 5/6.
} F5R6
CAN reception filter 5/6 (CAN_F5R6)
struct {
_BITS DATA: 8
CAN reception filter 5/7.
} F5R7
CAN reception filter 5/7 (CAN_F5R7)
struct {
_BITS DATA: 8

CAN reception filter 5/8.
} F5R8
CAN reception filter 5/8 (CAN_F5R8)
} PAGE_4
CAN page 4: Acceptance Filter 4:5 (CAN.PAGE_4)
struct {
struct {
_BITS TXRQ: 1
Transmission mailbox request.
_BITS ABRQ: 1
Abort request for mailbox.
_BITS RQCP: 1
Request completed.
_BITS TXOK: 1
Transmission OK.
_BITS ALST: 1
Arbitration lost.
_BITS TERR: 1
Transmission error.
_BITS __pad0__: 2
} MCSR
CAN message control/status register (CAN_MCSR)
struct {
_BITS DLC: 4
Data length code.
_BITS __pad0__: 3
_BITS TGT: 1
Transmit global time.
} MDLCR
CAN mailbox data length control register (CAN_MDLCR)
struct {
_BITS ID: 5
STID[10:6] or EXID[28:24].
_BITS RTR: 1
Remote transmission request.
_BITS IDE: 1
Extended identifier.
_BITS __pad0__: 1
} MIDR1
CAN mailbox identifier register 1 (CAN_MIDR1)
struct {
_BITS EXID: 2
EXID[17:16].
_BITS ID: 6
STID[5:0] or EXID[23:18].
} MIDR2
CAN mailbox identifier register 2 (CAN_MIDR2)
struct {
_BITS EXID: 8
EXID[15:8].
} MIDR3
CAN mailbox identifier register 3 (CAN_MIDR3)
struct {
_BITS EXID: 8
EXID[7:0].
} MIDR4
CAN mailbox identifier register 4 (CAN_MIDR4)
struct {

_BITS DATA: 8
data[7:0]
} MDAR1
CAN mailbox data register 1 (CAN_MDAR1)
struct {
_BITS DATA: 8
data[7:0]
} MDAR2
CAN mailbox data register 2 (CAN_MDAR2)
struct {
_BITS DATA: 8
data[7:0]
} MDAR3
CAN mailbox data register 3 (CAN_MDAR3)
struct {
_BITS DATA: 8
data[7:0]
} MDAR4
CAN mailbox data register 4 (CAN_MDAR4)
struct {
_BITS DATA: 8
data[7:0]
} MDAR5
CAN mailbox data register 5 (CAN_MDAR5)
struct {
_BITS DATA: 8
data[7:0]
} MDAR6
CAN mailbox data register 6 (CAN_MDAR6)
struct {
_BITS DATA: 8
data[7:0]
} MDAR7
CAN mailbox data register 7 (CAN_MDAR7)
struct {
_BITS DATA: 8
data[7:0]
} MDAR8
CAN mailbox data register 8 (CAN_MDAR8)
struct {
_BITS TIME: 8
Message time stamp [7:0].
} MTSRL
CAN mailbox time stamp register low byte (CAN_MTSRL)
struct {
_BITS TIME: 8
Message time stamp [15:8].
} MTSRH
CAN mailbox time stamp register high byte (CAN_MTSRH)
} PAGE_5
CAN page 5: Tx Mailbox 2 (CAN.PAGE_5)
struct {
struct {
_BITS EWGF: 1
Error warning flag.
_BITS EPVF: 1
Error passive flag.
_BITS BOFF: 1

Bus off flag.
_BITS __pad0__: 1
_BITS LEC: 3
Last error code.
_BITS __pad1__: 1
} ESR
CAN error status register (CAN_ESR)
struct {
_BITS EWGIE: 1
Error warning interrupt enable.
_BITS EPVIE: 1
Error passive interrupt enable.
_BITS BOFIE: 1
Bus-Off interrupt enable.
_BITS __pad0__: 1
_BITS LECIE: 1
Last error code interrupt enable.
_BITS __pad1__: 2
_BITS ERRIE: 1
Error interrupt enable.
} EIER
CAN error interrupt enable register (CAN_EIER)
struct {
_BITS TEC: 8
Transmit error counter.
} TECR
CAN transmit error counter register (CAN_TECR)
struct {
_BITS REC: 8
Receive error counter.
} RECR
CAN receive error counter register (CAN_RECR)
struct {
_BITS BRP: 6
Baud rate prescaler.
_BITS SJW: 2
Resynchronization jump width.
} BTR1
CAN bit timing register 1 (CAN_BTR1)
struct {
_BITS BS1: 4
Bit segment 1.
_BITS BS2: 3
Bit segment 2.
_BITS __pad0__: 1
} BTR2
CAN bit timing register 2 (CAN_BTR2)
uint8_t res [2]
Reserved registers (2B)
struct {
_BITS FML0: 1
Filter 0 mode low.
_BITS FMH0: 1
Filter 0 mode high.
_BITS FML1: 1
Filter 1 mode low.
_BITS FMH1: 1
Filter 1 mode high.

_BITS FML2: 1
Filter 2 mode low.
_BITS FMH2: 1
Filter 2 mode high.
_BITS FML3: 1
Filter 3 mode low.
_BITS FMH3: 1
Filter 3 mode high.
} FMR1
CAN filter mode register 1 (CAN_FMR1)
struct {
_BITS FML4: 1
Filter 4 mode low.
_BITS FMH4: 1
Filter 4 mode high.
_BITS FML5: 1
Filter 5 mode low.
_BITS FMH5: 1
Filter 5 mode high.
_BITS __pad0__: 4
} FMR2
CAN filter mode register 2 (CAN_FMR2)
struct {
_BITS FACT0: 1
Filter 0 active.
_BITS FSC0: 2
Filter 0 scale configuration.
_BITS __pad0__: 1
_BITS FACT1: 1
Filter 1 active.
_BITS FSC1: 2
Filter 1 scale configuration.
_BITS __pad1__: 1
} FCR1
CAN filter configuration register 1 (CAN_FCR1)
struct {
_BITS FACT2: 1
Filter 2 active.
_BITS FSC2: 2
Filter 2 scale configuration.
_BITS __pad0__: 1
_BITS FACT3: 1
Filter 3 active.
_BITS FSC3: 2
Filter 3 scale configuration.
_BITS __pad1__: 1
Reserve.
} FCR2
CAN filter configuration register 2 (CAN_FCR2)
struct {
_BITS FACT4: 1
Filter 4 active.
_BITS FSC4: 2
Filter 4 scale configuration.
_BITS __pad0__: 1
_BITS FACT5: 1
Filter 5 active.
_BITS FSC5: 2

Filter 5 scale configuration.
_BITS __pad1__: 1
Reserve.
} FCR3
CAN filter configuration register 3 (CAN_FCR3)
uint8_t res2 [3]
Reserved registers (3B)
} PAGE_6
CAN page 6: Configuration/Diagnostics (CAN.PAGE_6)
struct {
struct {
_BITS FMI: 8
Filter match index.
} MFMIR
CAN mailbox filter match index register (CAN_MFMIR)
struct {
_BITS DLC: 4
Data length code.
_BITS __pad0__: 3
_BITS TGT: 1
Transmit global time.
} MDLCR
CAN mailbox data length control register (CAN_MDLCR)
struct {
_BITS ID: 5
STID[10:6] or EXID[28:24].
_BITS RTR: 1
Remote transmission request.
_BITS IDE: 1
Extended identifier.
_BITS __pad0__: 1
} MIDR1
CAN mailbox identifier register 1 (CAN_MIDR1)
struct {
_BITS EXID: 2
EXID[17:16].
_BITS ID: 6
STID[5:0] or EXID[23:18].
} MIDR2
CAN mailbox identifier register 2 (CAN_MIDR2)
struct {
_BITS EXID: 8
EXID[15:8].
} MIDR3
CAN mailbox identifier register 3 (CAN_MIDR3)
struct {
_BITS EXID: 8
EXID[7:0].
} MIDR4
CAN mailbox identifier register 4 (CAN_MIDR4)
struct {
_BITS DATA: 8
CAN mailbox data register 1.
} MDAR1
CAN mailbox data register 1 (CAN_MDAR1)
struct {
_BITS DATA: 8
CAN mailbox data register 2.

} MDAR2
CAN mailbox data register 2 (CAN_MDAR2)
struct {
_BITS DATA: 8
CAN mailbox data register 3.
} MDAR3
CAN mailbox data register 3 (CAN_MDAR3)
struct {
_BITS DATA: 8
CAN mailbox data register 4.
} MDAR4
CAN mailbox data register 4 (CAN_MDAR4)
struct {
_BITS DATA: 8
CAN mailbox data register 5.
} MDAR5
CAN mailbox data register 5 (CAN_MDAR5)
struct {
_BITS DATA: 8
CAN mailbox data register 6.
} MDAR6
CAN mailbox data register 6 (CAN_MDAR6)
struct {
_BITS DATA: 8
CAN mailbox data register 7.
} MDAR7
CAN mailbox data register 7 (CAN_MDAR7)
struct {
_BITS DATA: 8
CAN mailbox data register 8.
} MDAR8
CAN mailbox data register 8 (CAN_MDAR8)
struct {
_BITS TIME: 8
Message time stamp [7:0].
} MTSRL
CAN mailbox time stamp register low byte (CAN_MTSRL)
struct {
_BITS TIME: 8
Message time stamp [15:8].
} MTSRH
CAN mailbox time stamp register high byte (CAN_MTSRH)
} PAGE_7
CAN page 7: Receive FIFO (CAN.PAGE_7)
} Page

paged CAN registers (selection via CAN_PSR)

6.5.1 Detailed Description

struct for controlling Controller Area Network Module (CAN)

Definition at line 5013 of file STM8AF_STM8S.h.

6.5.2 Field Documentation

6.5.2.1 __pad0__

_BITS __pad0__

Definition at line 5036 of file STM8AF_STM8S.h.

6.5.2.2 __pad1__

_BITS __pad1__

Reserve.

Definition at line 5049 of file STM8AF_STM8S.h.

6.5.2.3 ABOM

_BITS ABOM

Automatic Bus-Off Management.

Definition at line 5023 of file STM8AF_STM8S.h.

6.5.2.4 ABRQ

_BITS ABRQ

Abort request for mailbox.

Definition at line 5115 of file STM8AF_STM8S.h.

6.5.2.5 ALST

_BITS ALST

Arbitration lost.

Definition at line 5118 of file STM8AF_STM8S.h.

6.5.2.6 AWUM

_BITS AWUM

Automatic Wakeup Mode.

Definition at line 5022 of file STM8AF_STM8S.h.

6.5.2.7 BOFF

_BITS BOFF

Bus off flag.

Definition at line 5722 of file STM8AF_STM8S.h.

6.5.2.8 BOFIE

_BITS BOFIE

Bus-Off interrupt enable.

Definition at line 5733 of file STM8AF_STM8S.h.

6.5.2.9 BRP

_BITS BRP

Baud rate prescaler.

Definition at line 5755 of file STM8AF_STM8S.h.

6.5.2.10 BS1

_BITS BS1

Bit segment 1.

Definition at line 5762 of file STM8AF_STM8S.h.

6.5.2.11 BS2

_BITS BS2

Bit segment 2.

Definition at line 5763 of file STM8AF_STM8S.h.

6.5.2.12 BTR1

struct { ... } BTR1

CAN bit timing register 1 (CAN_BTR1)

6.5.2.13 BTR2

struct { ... } BTR2

CAN bit timing register 2 (CAN_BTR2)

6.5.2.14 CODE

_BITS CODE

Mailbox Code.

Definition at line 5055 of file STM8AF_STM8S.h.

6.5.2.15 DATA

_BITS DATA

data[7:0]

CAN mailbox data register 8.

CAN mailbox data register 7.

CAN mailbox data register 6.

CAN mailbox data register 5.

CAN mailbox data register 4.

CAN mailbox data register 3.

CAN mailbox data register 2.

CAN mailbox data register 1.

CAN reception filter 5/8.

CAN reception filter 5/7.

CAN reception filter 5/6.

CAN reception filter 5/5.

CAN reception filter 5/4.

CAN reception filter 5/3.

CAN reception filter 5/2.

CAN reception filter 5/1.

CAN reception filter 4/8.

CAN reception filter 4/7.

CAN reception filter 4/6.

CAN reception filter 4/5.

CAN reception filter 4/4.

CAN reception filter 4/3.

CAN reception filter 4/2.

CAN reception filter 4/1.

CAN reception filter 3/8.

CAN reception filter 3/7.

CAN reception filter 3/6.

CAN reception filter 3/5.

CAN reception filter 3/4.

CAN reception filter 3/3.

CAN reception filter 3/2.

CAN reception filter 3/1.

CAN reception filter 2/8.

CAN reception filter 2/7.

CAN reception filter 2/6.

CAN reception filter 2/5.

CAN reception filter 2/4.

CAN reception filter 2/3.

CAN reception filter 2/2.

CAN reception filter 2/1.

CAN reception filter 1/8.

CAN reception filter 1/7.

CAN reception filter 1/6.

CAN reception filter 1/5.

CAN reception filter 1/4.

CAN reception filter 1/3.

CAN reception filter 1/2.

CAN reception filter 1/1.

CAN reception filter 0/8.

CAN reception filter 0/7.

CAN reception filter 0/6.

CAN reception filter 0/5.

CAN reception filter 0/4.

CAN reception filter 0/3.

CAN reception filter 0/2.

CAN reception filter 0/1.

Definition at line 5162 of file STM8AF_STM8S.h.

6.5.2.16 DGR

```
struct { ...  } DGR
```

CAN diagnosis register (CAN_DGR)

6.5.2.17 DLC

```
_BITS DLC
```

Data length code.

Definition at line 5126 of file STM8AF_STM8S.h.

6.5.2.18 EIER

```
struct { ...  } EIER
```

CAN error interrupt enable register (CAN_EIER)

6.5.2.19 EPVF

```
_BITS EPVF
```

Error passive flag.

Definition at line 5721 of file STM8AF_STM8S.h.

6.5.2.20 EPVIE

```
_BITS EPVIE
```

Error passive interrupt enable.

Definition at line 5732 of file STM8AF_STM8S.h.

6.5.2.21 ERRI

_BITS ERRI

Error Interrupt.

Definition at line 5032 of file STM8AF_STM8S.h.

6.5.2.22 ERRIE

_BITS ERRIE

Error interrupt enable.

Definition at line 5737 of file STM8AF_STM8S.h.

6.5.2.23 ESR

struct { ... } ESR

CAN error status register (CAN_ESR)

6.5.2.24 EWGF

_BITS EWGF

Error warning flag.

Definition at line 5720 of file STM8AF_STM8S.h.

6.5.2.25 EWGIE

_BITS EWGIE

Error warning interrupt enable.

Definition at line 5731 of file STM8AF_STM8S.h.

6.5.2.26 EXID

_BITS EXID

EXID[17:16].

EXID[7:0].

EXID[15:8].

Definition at line 5143 of file STM8AF_STM8S.h.

6.5.2.27 F0R1

struct { ... } F0R1

CAN reception filter 0/1 (CAN_F0R1)

6.5.2.28 F0R2

struct { ... } F0R2

CAN reception filter 0/2 (CAN_F0R2)

6.5.2.29 F0R3

struct { ... } F0R3

CAN reception filter 0/3 (CAN_F0R3)

6.5.2.30 F0R4

struct { ... } F0R4

CAN reception filter 0/4 (CAN_F0R4)

6.5.2.31 F0R5

```
struct { ... } F0R5
```

CAN reception filter 0/5 (CAN_F0R5)

6.5.2.32 F0R6

```
struct { ... } F0R6
```

CAN reception filter 0/6 (CAN_F0R6)

6.5.2.33 F0R7

```
struct { ... } F0R7
```

CAN reception filter 0/7 (CAN_F0R7)

6.5.2.34 F0R8

```
struct { ... } F0R8
```

CAN reception filter 0/8 (CAN_F0R8)

6.5.2.35 F1R1

```
struct { ... } F1R1
```

CAN reception filter 1/1 (CAN_F1R1)

6.5.2.36 F1R2

```
struct { ... } F1R2
```

CAN reception filter 1/2 (CAN_F1R2)

6.5.2.37 F1R3

```
struct { ... } F1R3
```

CAN reception filter 1/3 (CAN_F1R3)

6.5.2.38 F1R4

```
struct { ... } F1R4
```

CAN reception filter 1/4 (CAN_F1R4)

6.5.2.39 F1R5

```
struct { ... } F1R5
```

CAN reception filter 1/5 (CAN_F1R5)

6.5.2.40 F1R6

```
struct { ... } F1R6
```

CAN reception filter 1/6 (CAN_F1R6)

6.5.2.41 F1R7

```
struct { ... } F1R7
```

CAN reception filter 1/7 (CAN_F1R7)

6.5.2.42 F1R8

```
struct { ... } F1R8
```

CAN reception filter 1/8 (CAN_F1R8)

6.5.2.43 F2R1

```
struct { ... } F2R1
```

CAN reception filter 2/1 (CAN_F2R1)

6.5.2.44 F2R2

```
struct { ... } F2R2
```

CAN reception filter 2/2 (CAN_F2R2)

6.5.2.45 F2R3

```
struct { ... } F2R3
```

CAN reception filter 2/3 (CAN_F2R3)

6.5.2.46 F2R4

```
struct { ... } F2R4
```

CAN reception filter 2/4 (CAN_F2R4)

6.5.2.47 F2R5

```
struct { ... } F2R5
```

CAN reception filter 2/5 (CAN_F2R5)

6.5.2.48 F2R6

```
struct { ... } F2R6
```

CAN reception filter 2/6 (CAN_F2R6)

6.5.2.49 F2R7

```
struct { ... } F2R7
```

CAN reception filter 2/7 (CAN_F2R7)

6.5.2.50 F2R8

```
struct { ... } F2R8
```

CAN reception filter 2/8 (CAN_F2R8)

6.5.2.51 F3R1

```
struct { ... } F3R1
```

CAN reception filter 3/1 (CAN_F3R1)

6.5.2.52 F3R2

```
struct { ... } F3R2
```

CAN reception filter 3/2 (CAN_F3R2)

6.5.2.53 F3R3

```
struct { ... } F3R3
```

CAN reception filter 3/3 (CAN_F3R3)

6.5.2.54 F3R4

```
struct { ... } F3R4
```

CAN reception filter 3/4 (CAN_F3R4)

6.5.2.55 F3R5

```
struct { ... } F3R5
```

CAN reception filter 3/5 (CAN_F3R5)

6.5.2.56 F3R6

```
struct { ... } F3R6
```

CAN reception filter 3/6 (CAN_F3R6)

6.5.2.57 F3R7

```
struct { ... } F3R7
```

CAN reception filter 3/7 (CAN_F3R7)

6.5.2.58 F3R8

```
struct { ... } F3R8
```

CAN reception filter 3/8 (CAN_F3R8)

6.5.2.59 F4R1

```
struct { ... } F4R1
```

CAN reception filter 4/1 (CAN_F4R1)

6.5.2.60 F4R2

```
struct { ... } F4R2
```

CAN reception filter 4/2 (CAN_F4R2)

6.5.2.61 F4R3

```
struct { ... } F4R3
```

CAN reception filter 4/3 (CAN_F4R3)

6.5.2.62 F4R4

```
struct { ... } F4R4
```

CAN reception filter 4/4 (CAN_F4R4)

6.5.2.63 F4R5

```
struct { ... } F4R5
```

CAN reception filter 4/5 (CAN_F4R5)

6.5.2.64 F4R6

```
struct { ... } F4R6
```

CAN reception filter 4/6 (CAN_F4R6)

6.5.2.65 F4R7

```
struct { ... } F4R7
```

CAN reception filter 4/7 (CAN_F4R7)

6.5.2.66 F4R8

```
struct { ... } F4R8
```

CAN reception filter 4/8 (CAN_F4R8)

6.5.2.67 F5R1

```
struct { ...  } F5R1
```

CAN reception filter 5/1 (CAN_F5R1)

6.5.2.68 F5R2

```
struct { ...  } F5R2
```

CAN reception filter 5/2 (CAN_F5R2)

6.5.2.69 F5R3

```
struct { ...  } F5R3
```

CAN reception filter 5/3 (CAN_F5R3)

6.5.2.70 F5R4

```
struct { ...  } F5R4
```

CAN reception filter 5/4 (CAN_F5R4)

6.5.2.71 F5R5

```
struct { ...  } F5R5
```

CAN reception filter 5/5 (CAN_F5R5)

6.5.2.72 F5R6

```
struct { ...  } F5R6
```

CAN reception filter 5/6 (CAN_F5R6)

6.5.2.73 F5R7

```
struct { ...  } F5R7
```

CAN reception filter 5/7 (CAN_F5R7)

6.5.2.74 F5R8

```
struct { ...  } F5R8
```

CAN reception filter 5/8 (CAN_F5R8)

6.5.2.75 FACT0

```
_BITS FACT0
```

Filter 0 active.

Definition at line 5797 of file STM8AF_STM8S.h.

6.5.2.76 FACT1

```
_BITS FACT1
```

Filter 1 active.

Definition at line 5800 of file STM8AF_STM8S.h.

6.5.2.77 FACT2

```
_BITS FACT2
```

Filter 2 active.

Definition at line 5808 of file STM8AF_STM8S.h.

6.5.2.78 FACT3

_BITS FACT3

Filter 3 active.

Definition at line 5811 of file STM8AF_STM8S.h.

6.5.2.79 FACT4

_BITS FACT4

Filter 4 active.

Definition at line 5819 of file STM8AF_STM8S.h.

6.5.2.80 FACT5

_BITS FACT5

Filter 5 active.

Definition at line 5822 of file STM8AF_STM8S.h.

6.5.2.81 FCR1

struct { ... } FCR1

CAN filter configuration register 1 (CAN_FCR1)

6.5.2.82 FCR2

struct { ... } FCR2

CAN filter configuration register 2 (CAN_FCR2)

6.5.2.83 FCR3

```
struct { ...  } FCR3
```

CAN filter configuration register 3 (CAN_FCR3)

6.5.2.84 FFIE

```
_BITS FFIE
```

FIFO Full Interrupt Enable.

Definition at line 5080 of file STM8AF_STM8S.h.

6.5.2.85 FMH0

```
_BITS FMH0
```

Filter 0 mode high.

Definition at line 5775 of file STM8AF_STM8S.h.

6.5.2.86 FMH1

```
_BITS FMH1
```

Filter 1 mode high.

Definition at line 5777 of file STM8AF_STM8S.h.

6.5.2.87 FMH2

```
_BITS FMH2
```

Filter 2 mode high.

Definition at line 5779 of file STM8AF_STM8S.h.

6.5.2.88 FMH3

_BITS FMH3

Filter 3 mode high.

Definition at line 5781 of file STM8AF_STM8S.h.

6.5.2.89 FMH4

_BITS FMH4

Filter 4 mode high.

Definition at line 5788 of file STM8AF_STM8S.h.

6.5.2.90 FMH5

_BITS FMH5

Filter 5 mode high.

Definition at line 5790 of file STM8AF_STM8S.h.

6.5.2.91 FMI

_BITS FMI

Filter match index.

Definition at line 5840 of file STM8AF_STM8S.h.

6.5.2.92 FML0

_BITS FML0

Filter 0 mode low.

Definition at line 5774 of file STM8AF_STM8S.h.

6.5.2.93 FML1

_BITS FML1

Filter 1 mode low.

Definition at line 5776 of file STM8AF_STM8S.h.

6.5.2.94 FML2

_BITS FML2

Filter 2 mode low.

Definition at line 5778 of file STM8AF_STM8S.h.

6.5.2.95 FML3

_BITS FML3

Filter 3 mode low.

Definition at line 5780 of file STM8AF_STM8S.h.

6.5.2.96 FML4

_BITS FML4

Filter 4 mode low.

Definition at line 5787 of file STM8AF_STM8S.h.

6.5.2.97 FML5

_BITS FML5

Filter 5 mode low.

Definition at line 5789 of file STM8AF_STM8S.h.

6.5.2.98 FMP

_BITS FMP

FIFO Message Pending.

Definition at line 5067 of file STM8AF_STM8S.h.

6.5.2.99 FMPIE

_BITS FMPIE

FIFO Message Pending Interrupt Enable.

Definition at line 5079 of file STM8AF_STM8S.h.

6.5.2.100 FMR1

struct { ... } FMR1

CAN filter mode register 1 (CAN_FMR1)

6.5.2.101 FMR2

struct { ... } FMR2

CAN filter mode register 2 (CAN_FMR2)

6.5.2.102 FOVIE

_BITS FOVIE

FIFO Overrun Interrupt Enable.

Definition at line 5081 of file STM8AF_STM8S.h.

6.5.2.103 FOVR

_BITS FOVR

FIFO Overrun.

Definition at line 5070 of file STM8AF_STM8S.h.

6.5.2.104 FSC0

_BITS FSC0

Filter 0 scale configuration.

Definition at line 5798 of file STM8AF_STM8S.h.

6.5.2.105 FSC1

_BITS FSC1

Filter 1 scale configuration.

Definition at line 5801 of file STM8AF_STM8S.h.

6.5.2.106 FSC2

_BITS FSC2

Filter 2 scale configuration.

Definition at line 5809 of file STM8AF_STM8S.h.

6.5.2.107 FSC3

_BITS FSC3

Filter 3 scale configuration.

Definition at line 5812 of file STM8AF_STM8S.h.

6.5.2.108 FSC4

_BITS FSC4

Filter 4 scale configuration.

Definition at line 5820 of file STM8AF_STM8S.h.

6.5.2.109 FSC5

_BITS FSC5

Filter 5 scale configuration.

Definition at line 5823 of file STM8AF_STM8S.h.

6.5.2.110 FULL

_BITS FULL

FIFO Full.

Definition at line 5069 of file STM8AF_STM8S.h.

6.5.2.111 ID

_BITS ID

STID[10:6] or EXID[28:24].

STID[5:0] or EXID[23:18].

Definition at line 5134 of file STM8AF_STM8S.h.

6.5.2.112 IDE

_BITS IDE

Extended identifier.

Definition at line 5136 of file STM8AF_STM8S.h.

6.5.2.113 IER

```
struct { ...  } IER
```

CAN interrupt enable register (CAN_IER)

6.5.2.114 INAK

```
_BITS INAK
```

Initialization Acknowledge.

Definition at line 5030 of file STM8AF_STM8S.h.

6.5.2.115 INRQ

```
_BITS INRQ
```

Initialization Request.

Definition at line 5017 of file STM8AF_STM8S.h.

6.5.2.116 LBKM

```
_BITS LBKM
```

Loop back mode.

Definition at line 5089 of file STM8AF_STM8S.h.

6.5.2.117 LEC

```
_BITS LEC
```

Last error code.

Definition at line 5724 of file STM8AF_STM8S.h.

6.5.2.118 LECIE

_BITS LECIE

Last error code interrupt enable.

Definition at line 5735 of file STM8AF_STM8S.h.

6.5.2.119 LOW0

_BITS LOW0

Lowest Priority Flag for Mailbox 0.

Definition at line 5059 of file STM8AF_STM8S.h.

6.5.2.120 LOW1

_BITS LOW1

Lowest Priority Flag for Mailbox 1.

Definition at line 5060 of file STM8AF_STM8S.h.

6.5.2.121 LOW2

_BITS LOW2

Lowest Priority Flag for Mailbox 2.

Definition at line 5061 of file STM8AF_STM8S.h.

6.5.2.122 MCR

struct { ... } MCR

CAN master control register (CAN_MCR)

6.5.2.123 MCSR [1/3]

```
struct { ... } MCSR
```

CAN message control/status register (CAN_MCSR)

6.5.2.124 MCSR [2/3]

```
struct { ... } MCSR
```

CAN message control/status register (CAN_MCSR)

6.5.2.125 MCSR [3/3]

```
struct { ... } MCSR
```

CAN message control/status register (CAN_MCSR)

6.5.2.126 MDAR1 [1/4]

```
struct { ... } MDAR1
```

CAN mailbox data register 1 (CAN_MDAR1)

6.5.2.127 MDAR1 [2/4]

```
struct { ... } MDAR1
```

CAN mailbox data register 1 (CAN_MDAR1)

6.5.2.128 MDAR1 [3/4]

```
struct { ... } MDAR1
```

CAN mailbox data register 1 (CAN_MDAR1)

6.5.2.129 MDAR1 [4/4]

```
struct { ... } MDAR1
```

CAN mailbox data register 1 (CAN_MDAR1)

6.5.2.130 MDAR2 [1/4]

```
struct { ... } MDAR2
```

CAN mailbox data register 2 (CAN_MDAR2)

6.5.2.131 MDAR2 [2/4]

```
struct { ... } MDAR2
```

CAN mailbox data register 2 (CAN_MDAR2)

6.5.2.132 MDAR2 [3/4]

```
struct { ... } MDAR2
```

CAN mailbox data register 2 (CAN_MDAR2)

6.5.2.133 MDAR2 [4/4]

```
struct { ... } MDAR2
```

CAN mailbox data register 2 (CAN_MDAR2)

6.5.2.134 MDAR3 [1/4]

```
struct { ... } MDAR3
```

CAN mailbox data register 3 (CAN_MDAR3)

6.5.2.135 MDAR3 [2/4]

```
struct { ... } MDAR3
```

CAN mailbox data register 3 (CAN_MDAR3)

6.5.2.136 MDAR3 [3/4]

```
struct { ... } MDAR3
```

CAN mailbox data register 3 (CAN_MDAR3)

6.5.2.137 MDAR3 [4/4]

```
struct { ... } MDAR3
```

CAN mailbox data register 3 (CAN_MDAR3)

6.5.2.138 MDAR4 [1/4]

```
struct { ... } MDAR4
```

CAN mailbox data register 4 (CAN_MDAR4)

6.5.2.139 MDAR4 [2/4]

```
struct { ... } MDAR4
```

CAN mailbox data register 4 (CAN_MDAR4)

6.5.2.140 MDAR4 [3/4]

```
struct { ... } MDAR4
```

CAN mailbox data register 4 (CAN_MDAR4)

6.5.2.141 MDAR4 [4/4]

```
struct { ... } MDAR4
```

CAN mailbox data register 4 (CAN_MDAR4)

6.5.2.142 MDAR5 [1/4]

```
struct { ... } MDAR5
```

CAN mailbox data register 5 (CAN_MDAR5)

6.5.2.143 MDAR5 [2/4]

```
struct { ... } MDAR5
```

CAN mailbox data register 5 (CAN_MDAR5)

6.5.2.144 MDAR5 [3/4]

```
struct { ... } MDAR5
```

CAN mailbox data register 5 (CAN_MDAR5)

6.5.2.145 MDAR5 [4/4]

```
struct { ... } MDAR5
```

CAN mailbox data register 5 (CAN_MDAR5)

6.5.2.146 MDAR6 [1/4]

```
struct { ... } MDAR6
```

CAN mailbox data register 6 (CAN_MDAR6)

6.5.2.147 MDAR6 [2/4]

```
struct { ... } MDAR6
```

CAN mailbox data register 6 (CAN_MDAR6)

6.5.2.148 MDAR6 [3/4]

```
struct { ... } MDAR6
```

CAN mailbox data register 6 (CAN_MDAR6)

6.5.2.149 MDAR6 [4/4]

```
struct { ... } MDAR6
```

CAN mailbox data register 6 (CAN_MDAR6)

6.5.2.150 MDAR7 [1/4]

```
struct { ... } MDAR7
```

CAN mailbox data register 7 (CAN_MDAR7)

6.5.2.151 MDAR7 [2/4]

```
struct { ... } MDAR7
```

CAN mailbox data register 7 (CAN_MDAR7)

6.5.2.152 MDAR7 [3/4]

```
struct { ... } MDAR7
```

CAN mailbox data register 7 (CAN_MDAR7)

6.5.2.153 MDAR7 [4/4]

```
struct { ... } MDAR7
```

CAN mailbox data register 7 (CAN_MDAR7)

6.5.2.154 MDAR8 [1/4]

```
struct { ... } MDAR8
```

CAN mailbox data register 8 (CAN_MDAR8)

6.5.2.155 MDAR8 [2/4]

```
struct { ... } MDAR8
```

CAN mailbox data register 8 (CAN_MDAR8)

6.5.2.156 MDAR8 [3/4]

```
struct { ... } MDAR8
```

CAN mailbox data register 8 (CAN_MDAR8)

6.5.2.157 MDAR8 [4/4]

```
struct { ... } MDAR8
```

CAN mailbox data register 8 (CAN_MDAR8)

6.5.2.158 MDLCR [1/4]

```
struct { ... } MDLCR
```

CAN mailbox data length control register (CAN_MDLCR)

6.5.2.159 MDLCR [2/4]

```
struct { ... } MDLCR
```

CAN mailbox data length control register (CAN_MDLCR)

6.5.2.160 MDLCR [3/4]

```
struct { ... } MDLCR
```

CAN mailbox data length control register (CAN_MDLCR)

6.5.2.161 MDLCR [4/4]

```
struct { ... } MDLCR
```

CAN mailbox data length control register (CAN_MDLCR)

6.5.2.162 MFMIR

```
struct { ... } MFMIR
```

CAN mailbox filter match index register (CAN_MFMIR)

6.5.2.163 MIDR1 [1/4]

```
struct { ... } MIDR1
```

CAN mailbox identifier register 1 (CAN_MIDR1)

6.5.2.164 MIDR1 [2/4]

```
struct { ... } MIDR1
```

CAN mailbox identifier register 1 (CAN_MIDR1)

6.5.2.165 MIDR1 [3/4]

```
struct { ... } MIDR1
```

CAN mailbox identifier register 1 (CAN_MIDR1)

6.5.2.166 MIDR1 [4/4]

```
struct { ... } MIDR1
```

CAN mailbox identifier register 1 (CAN_MIDR1)

6.5.2.167 MIDR2 [1/4]

```
struct { ... } MIDR2
```

CAN mailbox identifier register 2 (CAN_MIDR2)

6.5.2.168 MIDR2 [2/4]

```
struct { ... } MIDR2
```

CAN mailbox identifier register 2 (CAN_MIDR2)

6.5.2.169 MIDR2 [3/4]

```
struct { ... } MIDR2
```

CAN mailbox identifier register 2 (CAN_MIDR2)

6.5.2.170 MIDR2 [4/4]

```
struct { ... } MIDR2
```

CAN mailbox identifier register 2 (CAN_MIDR2)

6.5.2.171 MIDR3 [1/4]

```
struct { ... } MIDR3
```

CAN mailbox identifier register 3 (CAN_MIDR3)

6.5.2.172 MIDR3 [2/4]

```
struct { ... } MIDR3
```

CAN mailbox identifier register 3 (CAN_MIDR3)

6.5.2.173 MIDR3 [3/4]

```
struct { ... } MIDR3
```

CAN mailbox identifier register 3 (CAN_MIDR3)

6.5.2.174 MIDR3 [4/4]

```
struct { ... } MIDR3
```

CAN mailbox identifier register 3 (CAN_MIDR3)

6.5.2.175 MIDR4 [1/4]

```
struct { ... } MIDR4
```

CAN mailbox identifier register 4 (CAN_MIDR4)

6.5.2.176 MIDR4 [2/4]

```
struct { ... } MIDR4
```

CAN mailbox identifier register 4 (CAN_MIDR4)

6.5.2.177 MIDR4 [3/4]

```
struct { ... } MIDR4
```

CAN mailbox identifier register 4 (CAN_MIDR4)

6.5.2.178 MIDR4 [4/4]

```
struct { ... } MIDR4
```

CAN mailbox identifier register 4 (CAN_MIDR4)

6.5.2.179 MSR

```
struct { ... } MSR
```

CAN master status register (CAN_MSR)

6.5.2.180 MTSRH [1/4]

```
struct { ... } MTSRH
```

CAN mailbox time stamp register high byte (CAN_MTSRH)

6.5.2.181 MTSRH [2/4]

```
struct { ... } MTSRH
```

CAN mailbox time stamp register high byte (CAN_MTSRH)

6.5.2.182 MTSRH [3/4]

```
struct { ... } MTSRH
```

CAN mailbox time stamp register high byte (CAN_MTSRH)

6.5.2.183 MTSRH [4/4]

```
struct { ...  } MTSRH
```

CAN mailbox time stamp register high byte (CAN_MTSRH)

6.5.2.184 MTSRL [1/4]

```
struct { ...  } MTSRL
```

CAN mailbox time stamp register low byte (CAN_MTSRL)

6.5.2.185 MTSRL [2/4]

```
struct { ...  } MTSRL
```

CAN mailbox time stamp register low byte (CAN_MTSRL)

6.5.2.186 MTSRL [3/4]

```
struct { ...  } MTSRL
```

CAN mailbox time stamp register low byte (CAN_MTSRL)

6.5.2.187 MTSRL [4/4]

```
struct { ...  } MTSRL
```

CAN mailbox time stamp register low byte (CAN_MTSRL)

6.5.2.188 NART

```
_BITS NART
```

No Automatic Retransmission.

Definition at line 5021 of file STM8AF_STM8S.h.

6.5.2.189 Page

```
union { ... } Page
```

paged CAN registers (selection via CAN_PSR)

6.5.2.190 PAGE_0

```
struct { ... } PAGE_0
```

CAN page 0: Tx Mailbox 0 (CAN.PAGE_0)

6.5.2.191 PAGE_1

```
struct { ... } PAGE_1
```

CAN page 1: Tx Mailbox 1 (CAN.PAGE_1)

6.5.2.192 PAGE_2

```
struct { ... } PAGE_2
```

CAN page 2: Acceptance Filter 0:1 (CAN.PAGE_2)

6.5.2.193 PAGE_3

```
struct { ... } PAGE_3
```

CAN page 3: Acceptance Filter 2:3 (CAN.PAGE_3)

6.5.2.194 PAGE_4

```
struct { ... } PAGE_4
```

CAN page 4: Acceptance Filter 4:5 (CAN.PAGE_4)

6.5.2.195 PAGE_5

```
struct { ... } PAGE_5
```

CAN page 5: Tx Mailbox 2 (CAN.PAGE_5)

6.5.2.196 PAGE_6

```
struct { ... } PAGE_6
```

CAN page 6: Configuration/Diagnostics (CAN.PAGE_6)

6.5.2.197 PAGE_7

```
struct { ... } PAGE_7
```

CAN page 7: Receive FIFO (CAN.PAGE_7)

6.5.2.198 PS

```
_BITS PS
```

Page select.

Definition at line 5100 of file STM8AF_STM8S.h.

6.5.2.199 PSR

```
struct { ... } PSR
```

CAN page selection register for paged registers (CAN_PSR)

6.5.2.200 REC

```
_BITS REC
```

Receive error counter.

Definition at line 5749 of file STM8AF_STM8S.h.

6.5.2.201 RECR

```
struct { ...  } RECR
```

CAN receive error counter register (CAN_RECR)

6.5.2.202 res

```
uint8_t res[2]
```

Reserved registers (2B)

Definition at line 5769 of file STM8AF_STM8S.h.

6.5.2.203 res2

```
uint8_t res2[3]
```

Reserved registers (3B)

Definition at line 5829 of file STM8AF_STM8S.h.

6.5.2.204 RFLM

```
_BITS RFLM
```

Receive FIFO Locked Mode.

Definition at line 5020 of file STM8AF_STM8S.h.

6.5.2.205 RFOM

```
_BITS RFOM
```

Release FIFO Output Mailbox.

Definition at line 5071 of file STM8AF_STM8S.h.

6.5.2.206 RFR

```
struct { ...  } RFR
```

CAN receive FIFO register (CAN_RFR)

6.5.2.207 RQCP

```
_BITS RQCP
```

Request completed.

Definition at line 5116 of file STM8AF_STM8S.h.

6.5.2.208 RQCP0

```
_BITS RQCP0
```

Request Completed for Mailbox 0.

Definition at line 5042 of file STM8AF_STM8S.h.

6.5.2.209 RQCP1

```
_BITS RQCP1
```

Request Completed for Mailbox 1.

Definition at line 5043 of file STM8AF_STM8S.h.

6.5.2.210 RQCP2

```
_BITS RQCP2
```

Request Completed for Mailbox 2.

Definition at line 5044 of file STM8AF_STM8S.h.

6.5.2.211 RTR

_BITS RTR

Remote transmission request.

Definition at line 5135 of file STM8AF_STM8S.h.

6.5.2.212 RX

_BITS RX

Receive.

CAN Rx Signal.

Definition at line 5035 of file STM8AF_STM8S.h.

6.5.2.213 SAMP

_BITS SAMP

Last sample point.

Definition at line 5091 of file STM8AF_STM8S.h.

6.5.2.214 SILM

_BITS SILM

Silent mode.

Definition at line 5090 of file STM8AF_STM8S.h.

6.5.2.215 SJW

_BITS SJW

Resynchronization jump width.

Definition at line 5756 of file STM8AF_STM8S.h.

6.5.2.216 SLAK

_BITS SLAK

Sleep Acknowledge.

Definition at line 5031 of file STM8AF_STM8S.h.

6.5.2.217 SLEEP

_BITS SLEEP

Sleep Mode Request.

Definition at line 5018 of file STM8AF_STM8S.h.

6.5.2.218 TEC

_BITS TEC

Transmit error counter.

Definition at line 5743 of file STM8AF_STM8S.h.

6.5.2.219 TECR

struct { ... } TECR

CAN transmit error counter register (CAN_TECR)

6.5.2.220 TERR

_BITS TERR

Transmission error.

Definition at line 5119 of file STM8AF_STM8S.h.

6.5.2.221 TGT

_BITS TGT

Transmit global time.

Definition at line 5128 of file STM8AF_STM8S.h.

6.5.2.222 TIME

_BITS TIME

Message time stamp [7:0].

Message time stamp [15:8].

Definition at line 5210 of file STM8AF_STM8S.h.

6.5.2.223 TME0

_BITS TME0

Transmit Mailbox 0 Empty.

Definition at line 5056 of file STM8AF_STM8S.h.

6.5.2.224 TME1

_BITS TME1

Transmit Mailbox 1 Empty.

Definition at line 5057 of file STM8AF_STM8S.h.

6.5.2.225 TME2

_BITS TME2

Transmit Mailbox 2 Empty.

Definition at line 5058 of file STM8AF_STM8S.h.

6.5.2.226 TMEIE

_BITS TMEIE

Transmit Mailbox Empty Interrupt Enable.

Definition at line 5078 of file STM8AF_STM8S.h.

6.5.2.227 TPR

struct { ... } TPR

CAN transmit priority register (CAN_TPR)

6.5.2.228 TSR

struct { ... } TSR

CAN transmit status register (CAN_TSR)

6.5.2.229 TTCM

_BITS TTCM

Time Triggered Communication Mode.

Definition at line 5024 of file STM8AF_STM8S.h.

6.5.2.230 TX

_BITS TX

Transmit.

Definition at line 5034 of file STM8AF_STM8S.h.

6.5.2.231 TXFP

_BITS TXFP

Transmit FIFO Priority.

Definition at line 5019 of file STM8AF_STM8S.h.

6.5.2.232 TXM2E

_BITS TXM2E

TX Mailbox 2 enable.

Definition at line 5093 of file STM8AF_STM8S.h.

6.5.2.233 TXOK

_BITS TXOK

Transmission OK.

Definition at line 5117 of file STM8AF_STM8S.h.

6.5.2.234 TXOK0

_BITS TXOK0

Transmission ok for Mailbox 0.

Definition at line 5046 of file STM8AF_STM8S.h.

6.5.2.235 TXOK1

_BITS TXOK1

Transmission ok for Mailbox 1.

Definition at line 5047 of file STM8AF_STM8S.h.

6.5.2.236 TXOK2

_BITS TXOK2

Transmission ok for Mailbox 2.

Definition at line 5048 of file STM8AF_STM8S.h.

6.5.2.237 TXRQ

_BITS TXRQ

Transmission mailbox request.

Definition at line 5114 of file STM8AF_STM8S.h.

6.5.2.238 WKUI

_BITS WKUI

Wakeup Interrupt.

Definition at line 5033 of file STM8AF_STM8S.h.

6.5.2.239 WKUIE

_BITS WKUIE

Wakeup Interrupt Enable.

Definition at line 5083 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

6.6 CFG_t Struct Reference

struct for Global Configuration registers (CFG)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS SWD: 1
  SWIM disable.
  _BITS AL: 1
  Activation level.
  _BITS __pad0__: 6
  } GCR

  Global configuration register (CFG_GCR)
- struct {
  _BITS SWD: 1
  SWIM disable.
  _BITS AL: 1
  Activation level.
  _BITS __pad0__: 6
  } GCR

  Global configuration register (CFG_GCR)

6.6.1 Detailed Description

struct for Global Configuration registers (CFG)

Definition at line 6293 of file STM8AF_STM8S.h.

6.6.2 Field Documentation

6.6.2.1 __pad0__

```
_BITS __pad0__
```

Definition at line 6299 of file STM8AF_STM8S.h.

6.6.2.2 AL

```
_BITS AL
```

Activation level.

Definition at line 6298 of file STM8AF_STM8S.h.

6.6.2.3 GCR [1/2]

```
struct { ... } GCR
```

Global configuration register (CFG_GCR)

6.6.2.4 GCR [2/2]

```
struct { ... } GCR
```

Global configuration register (CFG_GCR)

6.6.2.5 SWD

```
_BITS SWD
```

SWIM disable.

Definition at line 6297 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.7 CLK_t Struct Reference

struct for configuring/monitoring clock module (CLK)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS HSIEN: 1
  High speed internal RC oscillator enable.
  _BITS HSIRDY: 1
  High speed internal oscillator ready flag.
  _BITS FHWU: 1
  Fast wakeup from Halt/Active-halt modes enable.
  _BITS LSIEN: 1
  Low speed internal RC oscillator enable.
  _BITS LSIRDY: 1
  Low speed internal oscillator ready flag.
  _BITS REGAH: 1
  Regulator power off in Active-halt mode enable.
  _BITS __pad0__: 2
  } ICKR

  Internal clock register (CLK_ICKR)
- struct {
  _BITS HSEEN: 1
  High speed external crystal oscillator enable.
  _BITS HSERDY: 1
  High speed external crystal oscillator ready.
  _BITS __pad0__: 6
  } ECKR

  External clock register (CLK_ECKR)
- uint8_t res [1]

  Reserved register (1B)
- struct {
  _BITS CKM: 8
  Clock master status bits.
  } CMSR

  Clock master status register (CLK_CMSR)
- struct {
  _BITS SWI: 8
  Clock master selection bits.
  } SWR

  Clock master switch register (CLK_SWR)
- struct {
  _BITS SWBSY: 1
  Switch busy flag.
  _BITS SWEN: 1
  Switch start/stop enable.
  _BITS SWIEN: 1
  Clock switch interrupt enable.
  _BITS SWIF: 1
  Clock switch interrupt flag.
  _BITS __pad0__: 4
  } SWCR

  Switch control register (CLK_SWCR)

- struct {
  _BITS CPUDIV: 3
  CPU clock prescaler.
  _BITS HSIDIV: 2
  High speed internal clock prescaler.
  _BITS __pad0__: 3
  } CKDIVR

  Clock divider register (CLK_CKDIVR)
- struct {
  _BITS PCKEN_I2C: 1
  clock enable I2C
  _BITS PCKEN_SPI: 1
  clock enable SPI
  _BITS PCKEN_UART1: 1
  clock enable UART1
  _BITS PCKEN_UART2: 1
  clock enable UART2
  _BITS PCKEN_TIM4_TIM6: 1
  clock enable TIM4/TIM6
  _BITS PCKEN_TIM2_TIM5: 1
  clock enable TIM4/TIM6
  _BITS PCKEN_TIM3: 1
  clock enable TIM3
  _BITS PCKEN_TIM1: 1
  clock enable TIM1
  } PCKENR1

  Peripheral clock gating register 1 (CLK_PCKENR1)
- struct {
  _BITS CSSEN: 1
  Clock security system enable.
  _BITS AUX: 1
  Auxiliary oscillator connected to master clock.
  _BITS CSSDIE: 1
  Clock security system detection interrupt enable.
  _BITS CSSD: 1
  Clock security system detection.
  _BITS __pad0__: 4
  } CSSR

  Clock security system register (CLK_CSSR)
- struct {
  _BITS CCOEN: 1
  Configurable clock output enable.
  _BITS CCOSEL: 4
  Configurable clock output selection.
  _BITS CCORDY: 1
  Configurable clock output ready.
  _BITS CCOBSY: 1
  Configurable clock output busy.
  _BITS __pad0__: 1
  } CCOR

  Configurable clock output register (CLK_CCOR)
- struct {
  _BITS __pad0__: 2

_BITS PCKEN_AWU: 1
clock enable AWU
_BITS PCKEN_ADC: 1
clock enable ADC
_BITS __pad1__: 3
_BITS PCKEN_CAN: 1
clock enable CAN
} PCKENR2

Peripheral clock gating register 2 (CLK_PCKENR2)

- uint8_t res2 [1]

Reserved register (1B). Was CAN clock control (obsolete as of STM8 UM rev 7)

- struct {
_BITS HSITRIM: 4
HSI trimming value (some devices only support 3 bits, see DS!)
_BITS __pad0__: 4
} HSITRIMR

HSI clock calibration trimming register (CLK_HSITRIMR)

- struct {
_BITS SWIMCLK: 1
SWIM clock divider.
_BITS __pad0__: 7
} SWIMCCR

SWIM clock control register (CLK_SWIMCCR)

- struct {
_BITS HSIDIV: 2
High speed internal clock prescaler.
_BITS __pad0__: 6
} CKDIVR

Internal clock register (CLK_ICKR)

- struct {
_BITS PCKEN_TIM2: 1
clock enable TIM2
_BITS PCKEN_TIM3: 1
clock enable TIM3
_BITS PCKEN_TIM4: 1
clock enable TIM4
_BITS PCKEN_I2C: 1
clock enable I2C
_BITS PCKEN_SPI: 1
clock enable SPI
_BITS PCKEN_USART: 1
clock enable USART
_BITS PCKEN_AWU_BEEP: 1
clock enable AWU/BEEP
_BITS __pad0__: 1
} PCKENR1

Peripheral clock gating register (CLK_PCKENR)

- struct {
_BITS CCOEN: 1
Configurable clock output enable.
_BITS CCOSEL: 2
Configurable clock output selection.

```
    _BITS __pad0__: 5
} CCOR
```

Configurable clock output register (CLK_CCOR)

6.7.1 Detailed Description

struct for configuring/monitoring clock module (CLK)

Definition at line 746 of file STM8AF_STM8S.h.

6.7.2 Field Documentation

6.7.2.1 __pad0__

`_BITS __pad0__`

Definition at line 756 of file STM8AF_STM8S.h.

6.7.2.2 __pad1__

`_BITS __pad1__`

Definition at line 840 of file STM8AF_STM8S.h.

6.7.2.3 AUX

`_BITS AUX`

Auxiliary oscillator connected to master clock.

Definition at line 818 of file STM8AF_STM8S.h.

6.7.2.4 CCOBSY

`_BITS CCOBSY`

Configurable clock output busy.

Definition at line 830 of file STM8AF_STM8S.h.

6.7.2.5 CCOEN

_BITS CCOEN

Configurable clock output enable.

Definition at line 827 of file STM8AF_STM8S.h.

6.7.2.6 CCOR [1/2]

struct { ... } CCOR

Configurable clock output register (CLK_CCOR)

6.7.2.7 CCOR [2/2]

struct { ... } CCOR

Configurable clock output register (CLK_CCOR)

6.7.2.8 CCORDY

_BITS CCORDY

Configurable clock output ready.

Definition at line 829 of file STM8AF_STM8S.h.

6.7.2.9 CCOSEL

_BITS CCOSEL

Configurable clock output selection.

Definition at line 828 of file STM8AF_STM8S.h.

6.7.2.10 CKDIVR [1/2]

```
struct { ... } CKDIVR
```

Internal clock register (CLK_ICKR)

6.7.2.11 CKDIVR [2/2]

```
struct { ... } CKDIVR
```

Clock divider register (CLK_CKDIVR)

6.7.2.12 CKM

```
_BITS CKM
```

Clock master status bits.

Definition at line 774 of file STM8AF_STM8S.h.

6.7.2.13 CMSR

```
struct { ... } CMSR
```

Clock master status register (CLK_CMSR)

6.7.2.14 CPUDIV

```
_BITS CPUDIV
```

CPU clock prescaler.

Definition at line 796 of file STM8AF_STM8S.h.

6.7.2.15 CSSD

```
_BITS CSSD
```

Clock security system detection.

Definition at line 820 of file STM8AF_STM8S.h.

6.7.2.16 CSSDIE

_BITS CSSDIE

Clock security system detection interrupt enable.

Definition at line 819 of file STM8AF_STM8S.h.

6.7.2.17 CSSEN

_BITS CSSEN

Clock security system enable.

Definition at line 817 of file STM8AF_STM8S.h.

6.7.2.18 CSSR

struct { ... } CSSR

Clock security system register (CLK_CSSR)

6.7.2.19 ECKR

struct { ... } ECKR

External clock register (CLK_ECKR)

6.7.2.20 FHWU

_BITS FHWU

Fast wakeup from Halt/Active-halt modes enable.

Definition at line 752 of file STM8AF_STM8S.h.

6.7.2.21 HSEEN

_BITS HSEEN

High speed external crystal oscillator enable.

Definition at line 762 of file STM8AF_STM8S.h.

6.7.2.22 HSERDY

_BITS HSERDY

High speed external crystal oscillator ready.

Definition at line 763 of file STM8AF_STM8S.h.

6.7.2.23 HSIDIV

_BITS HSIDIV

High speed internal clock prescaler.

Definition at line 797 of file STM8AF_STM8S.h.

6.7.2.24 HSIEN

_BITS HSIEN

High speed internal RC oscillator enable.

Definition at line 750 of file STM8AF_STM8S.h.

6.7.2.25 HSIRDY

_BITS HSIRDY

High speed internal oscillator ready flag.

Definition at line 751 of file STM8AF_STM8S.h.

6.7.2.26 HSITRIM

```
_BITS HSITRIM
```

HSI trimming value (some devices only support 3 bits, see DS!)

Definition at line 851 of file STM8AF_STM8S.h.

6.7.2.27 HSITRIMR

```
struct { ...  } HSITRIMR
```

HSI clock calibration trimming register (CLK_HSITRIMR)

6.7.2.28 ICKR

```
struct { ...  } ICKR
```

Internal clock register (CLK_ICKR)

6.7.2.29 LSIEN

```
_BITS LSIEN
```

Low speed internal RC oscillator enable.

Definition at line 753 of file STM8AF_STM8S.h.

6.7.2.30 LSIRDY

```
_BITS LSIRDY
```

Low speed internal oscillator ready flag.

Definition at line 754 of file STM8AF_STM8S.h.

6.7.2.31 PCKEN_ADC

_BITS PCKEN_ADC

clock enable ADC

Definition at line 839 of file STM8AF_STM8S.h.

6.7.2.32 PCKEN_AWU

_BITS PCKEN_AWU

clock enable AWU

Definition at line 838 of file STM8AF_STM8S.h.

6.7.2.33 PCKEN_AWU_BEEP

_BITS PCKEN_AWU_BEEP

clock enable AWU/BEEP

Definition at line 712 of file STM8L10x.h.

6.7.2.34 PCKEN_CAN

_BITS PCKEN_CAN

clock enable CAN

Definition at line 841 of file STM8AF_STM8S.h.

6.7.2.35 PCKEN_I2C

_BITS PCKEN_I2C

clock enable I2C

Definition at line 804 of file STM8AF_STM8S.h.

6.7.2.36 PCKEN_SPI

_BITS PCKEN_SPI

clock enable SPI

Definition at line 805 of file STM8AF_STM8S.h.

6.7.2.37 PCKEN_TIM1

_BITS PCKEN_TIM1

clock enable TIM1

Definition at line 811 of file STM8AF_STM8S.h.

6.7.2.38 PCKEN_TIM2

_BITS PCKEN_TIM2

clock enable TIM2

Definition at line 706 of file STM8L10x.h.

6.7.2.39 PCKEN_TIM2_TIM5

_BITS PCKEN_TIM2_TIM5

clock enable TIM4/TIM6

Definition at line 809 of file STM8AF_STM8S.h.

6.7.2.40 PCKEN_TIM3

_BITS PCKEN_TIM3

clock enable TIM3

Definition at line 810 of file STM8AF_STM8S.h.

6.7.2.41 PCKEN_TIM4

_BITS PCKEN_TIM4

clock enable TIM4

Definition at line 708 of file STM8L10x.h.

6.7.2.42 PCKEN_TIM4_TIM6

_BITS PCKEN_TIM4_TIM6

clock enable TIM4/TIM6

Definition at line 808 of file STM8AF_STM8S.h.

6.7.2.43 PCKEN_UART1

_BITS PCKEN_UART1

clock enable UART1

Definition at line 806 of file STM8AF_STM8S.h.

6.7.2.44 PCKEN_UART2

_BITS PCKEN_UART2

clock enable UART2

Definition at line 807 of file STM8AF_STM8S.h.

6.7.2.45 PCKEN_USART

_BITS PCKEN_USART

clock enable USART

Definition at line 711 of file STM8L10x.h.

6.7.2.46 PCKENR1 [1/2]

```
struct { ...  } PCKENR1
```

Peripheral clock gating register (CLK_PCKENR)

6.7.2.47 PCKENR1 [2/2]

```
struct { ...  } PCKENR1
```

Peripheral clock gating register 1 (CLK_PCKENR1)

6.7.2.48 PCKENR2

```
struct { ...  } PCKENR2
```

Peripheral clock gating register 2 (CLK_PCKENR2)

6.7.2.49 REGAH

```
_BITS REGAH
```

Regulator power off in Active-halt mode enable.

Definition at line 755 of file STM8AF_STM8S.h.

6.7.2.50 res

```
uint8_t res
```

Reserved register (1B)

Reserved register (2B)

Definition at line 769 of file STM8AF_STM8S.h.

6.7.2.51 res2

```
uint8_t res2
```

Reserved register (1B). Was CAN clock control (obsolete as of STM8 UM rev 7)

Reserved register (1B)

Definition at line 846 of file STM8AF_STM8S.h.

6.7.2.52 SWBSY

```
_BITS SWBSY
```

Switch busy flag.

Definition at line 786 of file STM8AF_STM8S.h.

6.7.2.53 SWCR

```
struct { ...  } SWCR
```

Switch control register (CLK_SWCR)

6.7.2.54 SWEN

```
_BITS SWEN
```

Switch start/stop enable.

Definition at line 787 of file STM8AF_STM8S.h.

6.7.2.55 SWI

```
_BITS SWI
```

Clock master selection bits.

Definition at line 780 of file STM8AF_STM8S.h.

6.7.2.56 SWIEN

_BITS SWIEN

Clock switch interrupt enable.

Definition at line 788 of file STM8AF_STM8S.h.

6.7.2.57 SWIF

_BITS SWIF

Clock switch interrupt flag.

Definition at line 789 of file STM8AF_STM8S.h.

6.7.2.58 SWIMCCR

struct { ... } SWIMCCR

SWIM clock control register (CLK_SWIMCCR)

6.7.2.59 SWIMCLK

_BITS SWIMCLK

SWIM clock divider.

Definition at line 858 of file STM8AF_STM8S.h.

6.7.2.60 SWR

struct { ... } SWR

Clock master switch register (CLK_SWR)

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.8 COMP_t Struct Reference

struct for Comparator Module (COMP)

```
#include <STM8L10x.h>
```

Data Fields

- struct {
  _BITS BIAS_EN: 1
  Bias enable.
  _BITS COMP1_EN: 1
  First comparator enable.
  _BITS COMP2_EN: 1
  Second comparator enable.
  _BITS COMPREF: 1
  Comparator reference.
  _BITS POL: 1
  Comparator polarity.
  _BITS CNF_TIM: 2
  Comparator 1/2 output connected to TIM2/3 capture or break.
  _BITS IC1_BK: 1
  Input capture 1 / break selection.
  } CR

  Comparator control register (COMP_CR)
- struct {
  _BITS COMP1_OUT: 1
  First comparator output.
  _BITS COMP2_OUT: 1
  Second comparator output.
  _BITS __pad0__: 2
  _BITS CEF1: 1
  First comparator event flag.
  _BITS ITEN1: 1
  First comparator interrupt enable.
  _BITS CEF2: 1
  Second comparator event flag.
  _BITS ITEN2: 1
  Second comparator interrupt enable.
  } CSR

  Comparator control status register (COMP_CSR)
- struct {
  _BITS COMP1_CH1: 1
  Comparator 1 switch 1 enable.
  _BITS COMP1_CH2: 1
  Comparator 1 switch 2 enable.
  _BITS COMP1_CH3: 1
  Comparator 1 switch 3 enable.
  _BITS COMP1_CH4: 1
  Comparator 1 switch 4 enable.
  _BITS COMP2_CH1: 1
  Comparator 2 switch 1 enable.
  _BITS COMP2_CH2: 1

Comparator 2 switch 2 enable.
_BITS COMP2_CH3: 1
Comparator 2 switch 3 enable.
_BITS COMP2_CH4: 1
Comparator 2 switch 4 enable.
} CCS

Comparator channel selection (COMP_CCS)

6.8.1 Detailed Description

struct for Comparator Module (COMP)

Definition at line 2378 of file STM8L10x.h.

6.8.2 Field Documentation

6.8.2.1 __pad0__

_BITS __pad0__

Definition at line 2395 of file STM8L10x.h.

6.8.2.2 BIAS_EN

_BITS BIAS_EN

Bias enable.

Definition at line 2382 of file STM8L10x.h.

6.8.2.3 CCS

struct { ... } CCS

Comparator channel selection (COMP_CCS)

6.8.2.4 CEF1

_BITS CEF1

First comparator event flag.

Definition at line 2396 of file STM8L10x.h.

6.8.2.5 CEF2

_BITS CEF2

Second comparator event flag.

Definition at line 2398 of file STM8L10x.h.

6.8.2.6 CNF_TIM

_BITS CNF_TIM

Comparator 1/2 output connected to TIM2/3 capture or break.

Definition at line 2387 of file STM8L10x.h.

6.8.2.7 COMP1_CH1

_BITS COMP1_CH1

Comparator 1 switch 1 enable.

Definition at line 2404 of file STM8L10x.h.

6.8.2.8 COMP1_CH2

_BITS COMP1_CH2

Comparator 1 switch 2 enable.

Definition at line 2405 of file STM8L10x.h.

6.8.2.9 COMP1_CH3

_BITS COMP1_CH3

Comparator 1 switch 3 enable.

Definition at line 2406 of file STM8L10x.h.

6.8.2.10 COMP1_CH4

_BITS COMP1_CH4

Comparator 1 switch 4 enable.

Definition at line 2407 of file STM8L10x.h.

6.8.2.11 COMP1_EN

_BITS COMP1_EN

First comparator enable.

Definition at line 2383 of file STM8L10x.h.

6.8.2.12 COMP1_OUT

_BITS COMP1_OUT

First comparator output.

Definition at line 2393 of file STM8L10x.h.

6.8.2.13 COMP2_CH1

_BITS COMP2_CH1

Comparator 2 switch 1 enable.

Definition at line 2408 of file STM8L10x.h.

6.8.2.14 COMP2_CH2

_BITS COMP2_CH2

Comparator 2 switch 2 enable.

Definition at line 2409 of file STM8L10x.h.

6.8.2.15 COMP2_CH3

_BITS COMP2_CH3

Comparator 2 switch 3 enable.

Definition at line 2410 of file STM8L10x.h.

6.8.2.16 COMP2_CH4

_BITS COMP2_CH4

Comparator 2 switch 4 enable.

Definition at line 2411 of file STM8L10x.h.

6.8.2.17 COMP2_EN

_BITS COMP2_EN

Second comparator enable.

Definition at line 2384 of file STM8L10x.h.

6.8.2.18 COMP2_OUT

_BITS COMP2_OUT

Second comparator output.

Definition at line 2394 of file STM8L10x.h.

6.8.2.19 COMPREF

_BITS COMPREF

Comparator reference.

Definition at line 2385 of file STM8L10x.h.

6.8.2.20 CR

struct { ... } CR

Comparator control register (COMP_CR)

6.8.2.21 CSR

struct { ... } CSR

Comparator control status register (COMP_CSR)

6.8.2.22 IC1_BK

_BITS IC1_BK

Input capture 1 / break selection.

Definition at line 2388 of file STM8L10x.h.

6.8.2.23 ITEN1

_BITS ITEN1

First comparator interrupt enable.

Definition at line 2397 of file STM8L10x.h.

6.8.2.24 ITEN2

_BITS ITEN2

Second comparator interrupt enable.

Definition at line 2399 of file STM8L10x.h.

6.8.2.25 POL

_BITS POL

Comparator polarity.

Definition at line 2386 of file STM8L10x.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.9 EXTI_t Struct Reference

struct for configuring external port interrupts (EXTI)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS PAIS: 2
  Port A external interrupt sensitivity bits.
  _BITS PBIS: 2
  Port B external interrupt sensitivity bits.
  _BITS PCIS: 2
  Port C external interrupt sensitivity bits.
  _BITS PDIS: 2
  Port D external interrupt sensitivity bits.
  } CR1

  External interrupt control register 1 (EXTI_CR1)
- struct {
  _BITS PEIS: 2
  Port E external interrupt sensitivity bits.
  _BITS TLIS: 1
  Top level interrupt sensitivity.
  _BITS __pad0__: 5
  } CR2

  External interrupt control register 2 (EXTI_CR2)

- struct {
  _BITS P0IS: 2
  Portx bit 0 external interrupt sensitivity bits.
  _BITS P1IS: 2
  Portx bit 1 external interrupt sensitivity bits.
  _BITS P2IS: 2
  Portx bit 2 external interrupt sensitivity bits.
  _BITS P3IS: 2
  Portx bit 3 external interrupt sensitivity bits.
  } CR1

External interrupt control register 1 (EXTI_CR1)

- struct {
  _BITS P4IS: 2
  Portx bit 4 external interrupt sensitivity bits.
  _BITS P5IS: 2
  Portx bit 5 external interrupt sensitivity bits.
  _BITS P6IS: 2
  Portx bit 6 external interrupt sensitivity bits.
  _BITS P7IS: 2
  Portx bit 7 external interrupt sensitivity bits.
  } CR2

External interrupt control register 2 (EXTI_CR2)

- struct {
  _BITS PBIS: 2
  Port B external interrupt sensitivity bits.
  _BITS PDIS: 2
  Port D external interrupt sensitivity bits.
  _BITS __pad0__: 4
  } CR3

External interrupt control register 3 (EXTI_CR3)

- struct {
  _BITS P0F: 1
  Portx bit 0 external interrupt flag.
  _BITS P1F: 1
  Portx bit 1 external interrupt flag.
  _BITS P2F: 1
  Portx bit 2 external interrupt flag.
  _BITS P3F: 1
  Portx bit 3 external interrupt flag.
  _BITS P4F: 1
  Portx bit 4 external interrupt flag.
  _BITS P5F: 1
  Portx bit 5 external interrupt flag.
  _BITS P6F: 1
  Portx bit 6 external interrupt flag.
  _BITS P7F: 1
  Portx bit 7 external interrupt flag.
  } SR1

External interrupt status register 1 (EXTI_SR1)

- struct {
  _BITS PBF: 1
  Port B external interrupt flag.
  _BITS PDF: 1

Port D external interrupt flag.
_BITS __pad0__: 6
} SR2

External interrupt status register 2 (EXTI_SR2)

- struct {
_BITS PBLIS: 1
Port B, pins 0..3 external interrupt select.
_BITS PBHIS: 1
Port B, pins 4..7 external interrupt select.
_BITS PDLIS: 1
Port D, pins 0..3 external interrupt select.
_BITS PDHIS: 1
Port D, pins 4..7 external interrupt select.
_BITS __pad0__: 4
} CONF

External interrupt port selector (EXTI_CONF)

6.9.1 Detailed Description

struct for configuring external port interrupts (EXTI)

Definition at line 650 of file STM8AF_STM8S.h.

6.9.2 Field Documentation

6.9.2.1 __pad0__

`_BITS __pad0__`

Definition at line 665 of file STM8AF_STM8S.h.

6.9.2.2 CONF

`struct { ... } CONF`

External interrupt port selector (EXTI_CONF)

6.9.2.3 CR1 [1/2]

`struct { ... } CR1`

External interrupt control register 1 (EXTI_CR1)

6.9.2.4 CR1 [2/2]

```
struct { ...  } CR1
```

External interrupt control register 1 (EXTI_CR1)

6.9.2.5 CR2 [1/2]

```
struct { ...  } CR2
```

External interrupt control register 2 (EXTI_CR2)

6.9.2.6 CR2 [2/2]

```
struct { ...  } CR2
```

External interrupt control register 2 (EXTI_CR2)

6.9.2.7 CR3

```
struct { ...  } CR3
```

External interrupt control register 3 (EXTI_CR3)

6.9.2.8 P0F

```
_BITS P0F
```

Portx bit 0 external interrupt flag.

Definition at line 534 of file STM8L10x.h.

6.9.2.9 P0IS

```
_BITS P0IS
```

Portx bit 0 external interrupt sensitivity bits.

Definition at line 508 of file STM8L10x.h.

6.9.2.10 P1F

_BITS P1F

Portx bit 1 external interrupt flag.

Definition at line 535 of file STM8L10x.h.

6.9.2.11 P1IS

_BITS P1IS

Portx bit 1 external interrupt sensitivity bits.

Definition at line 509 of file STM8L10x.h.

6.9.2.12 P2F

_BITS P2F

Portx bit 2 external interrupt flag.

Definition at line 536 of file STM8L10x.h.

6.9.2.13 P2IS

_BITS P2IS

Portx bit 2 external interrupt sensitivity bits.

Definition at line 510 of file STM8L10x.h.

6.9.2.14 P3F

_BITS P3F

Portx bit 3 external interrupt flag.

Definition at line 537 of file STM8L10x.h.

6.9.2.15 P3IS

_BITS P3IS

Portx bit 3 external interrupt sensitivity bits.

Definition at line 511 of file STM8L10x.h.

6.9.2.16 P4F

_BITS P4F

Portx bit 4 external interrupt flag.

Definition at line 538 of file STM8L10x.h.

6.9.2.17 P4IS

_BITS P4IS

Portx bit 4 external interrupt sensitivity bits.

Definition at line 517 of file STM8L10x.h.

6.9.2.18 P5F

_BITS P5F

Portx bit 5 external interrupt flag.

Definition at line 539 of file STM8L10x.h.

6.9.2.19 P5IS

_BITS P5IS

Portx bit 5 external interrupt sensitivity bits.

Definition at line 518 of file STM8L10x.h.

6.9.2.20 P6F

_BITS P6F

Portx bit 6 external interrupt flag.

Definition at line 540 of file STM8L10x.h.

6.9.2.21 P6IS

_BITS P6IS

Portx bit 6 external interrupt sensitivity bits.

Definition at line 519 of file STM8L10x.h.

6.9.2.22 P7F

_BITS P7F

Portx bit 7 external interrupt flag.

Definition at line 541 of file STM8L10x.h.

6.9.2.23 P7IS

_BITS P7IS

Portx bit 7 external interrupt sensitivity bits.

Definition at line 520 of file STM8L10x.h.

6.9.2.24 PAIS

_BITS PAIS

Port A external interrupt sensitivity bits.

Definition at line 654 of file STM8AF_STM8S.h.

6.9.2.25 PBF

_BITS PBF

Port B external interrupt flag.

Definition at line 547 of file STM8L10x.h.

6.9.2.26 PBHIS

_BITS PBHIS

Port B, pins 4..7 external interrupt select.

Definition at line 556 of file STM8L10x.h.

6.9.2.27 PBIS

_BITS PBIS

Port B external interrupt sensitivity bits.

Definition at line 655 of file STM8AF_STM8S.h.

6.9.2.28 PBLIS

_BITS PBLIS

Port B, pins 0..3 external interrupt select.

Definition at line 555 of file STM8L10x.h.

6.9.2.29 PCIS

_BITS PCIS

Port C external interrupt sensitivity bits.

Definition at line 656 of file STM8AF_STM8S.h.

6.9.2.30 PDF

_BITS PDF

Port D external interrupt flag.

Definition at line 548 of file STM8L10x.h.

6.9.2.31 PDHIS

_BITS PDHIS

Port D, pins 4..7 external interrupt select.

Definition at line 558 of file STM8L10x.h.

6.9.2.32 PDIS

_BITS PDIS

Port D external interrupt sensitivity bits.

Definition at line 657 of file STM8AF_STM8S.h.

6.9.2.33 PDLIS

_BITS PDLIS

Port D, pins 0..3 external interrupt select.

Definition at line 557 of file STM8L10x.h.

6.9.2.34 PEIS

_BITS PEIS

Port E external interrupt sensitivity bits.

Definition at line 663 of file STM8AF_STM8S.h.

6.9.2.35 SR1

```
struct { ...  } SR1
```

External interrupt status register 1 (EXTI_SR1)

6.9.2.36 SR2

```
struct { ...  } SR2
```

External interrupt status register 2 (EXTI_SR2)

6.9.2.37 TLIS

```
_BITS TLIS
```

Top level interrupt sensitivity.

Definition at line 664 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.10 FLASH_t Struct Reference

struct to control write/erase of flash memory (FLASH)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS FIX: 1
  Fixed Byte programming time.
  _BITS IE: 1
  Flash Interrupt enable.
  _BITS AHALT: 1
  Power-down in Active-halt mode.
  _BITS HALT: 1
  Power-down in Halt mode.
  _BITS __pad0__: 4
  } CR1

  Flash control register 1 (FLASH_CR1)
- struct {
  _BITS PRG: 1
  Standard block programming.
  _BITS __pad0__: 3
  _BITS FPRG: 1
  Fast block programming.
  _BITS ERASE: 1
  Block erasing.
  _BITS WPRG: 1
  Word programming.
  _BITS OPT: 1
  Write option bytes.
  } CR2

  Flash control register 2 (FLASH_CR2)
- struct {
  _BITS NPRG: 1
  Standard block programming.
  _BITS __pad0__: 3
  _BITS NFPRG: 1
  Fast block programming.
  _BITS NERASE: 1
  Block erasing.
  _BITS NWPRG: 1
  Word programming.
  _BITS NOPT: 1
  Write option bytes.
  } NCR2

  Complementary flash control register 2 (FLASH_NCR2)
- struct {
  _BITS WPB: 6
  User boot code area protection bits.
  _BITS __pad0__: 2
  } FPR

  Flash protection register (FLASH_FPR)
- struct {
  _BITS NWPB: 6
  User boot code area protection bits.
  _BITS __pad0__: 2
  } NFPR

Complementary flash protection register (FLASH_NFPR)

- struct {
  _BITS WR_PG_DIS: 1
  Write attempted to protected page flag.
  _BITS PUL: 1
  Flash Program memory unlocked flag.
  _BITS EOP: 1
  End of programming (write or erase operation) flag.
  _BITS DUL: 1
  Data EEPROM area unlocked flag.
  _BITS __pad0__: 2
  _BITS HVOFF: 1
  End of high voltage flag.
  _BITS __pad1__: 1
  } IAPSR

Flash status register (FLASH_IAPSR)

- uint8_t res [2]

Reserved registers (2B)

- struct {
  _BITS PUK: 8
  Program memory write unlock key.
  } PUKR

Flash program memory unprotecting key register (FLASH_PUKR)

- uint8_t res2 [1]

Reserved register (1B)

- struct {
  _BITS DUK: 8
  Data EEPROM write unlock key.
  } DUKR

Data EEPROM unprotection key register (FLASH_DUKR)

- struct {
  _BITS FIX: 1
  Fixed Byte programming time.
  _BITS IE: 1
  Flash Interrupt enable.
  _BITS __pad0__: 6
  } CR1

Flash control register 1 (FLASH_CR1)

- struct {
  _BITS PRG: 1
  Standard block programming.
  _BITS __pad0__: 3
  _BITS FPRG: 1
  Fast block programming.
  _BITS ERASE: 1
  Block erasing.
  _BITS WPRG: 1
  Word programming.
  _BITS OPT: 1
  Write option bytes.
  } CR2

Flash control register 2 (FLASH_CR2)

- struct {
  _BITS PUK: 8
  Program memory write unlock key.
  } PUKR

Flash program memory unprotecting key register (FLASH_PUKR)

- struct {
  _BITS DUK: 8
  Data EEPROM write unlock key.
  } DUKR

Data EEPROM unprotection key register (FLASH_DUKR)

- struct {
  _BITS WR_PG_DIS: 1
  Write attempted to protected page flag.
  _BITS PUL: 1
  Flash Program memory unlocked flag.
  _BITS EOP: 1
  End of programming (write or erase operation) flag.
  _BITS DUL: 1
  Data EEPROM area unlocked flag.
  _BITS __pad0__: 4
  } IAPSR

Flash status register (FLASH_IAPSR)

6.10.1 Detailed Description

struct to control write/erase of flash memory (FLASH)

Definition at line 504 of file STM8AF_STM8S.h.

6.10.2 Field Documentation

6.10.2.1 __pad0__

```
_BITS __pad0__
```

Definition at line 512 of file STM8AF_STM8S.h.

6.10.2.2 __pad1__

```
_BITS __pad1__
```

Definition at line 560 of file STM8AF_STM8S.h.

6.10.2.3 AHALT

_BITS AHALT

Power-down in Active-halt mode.

Definition at line 510 of file STM8AF_STM8S.h.

6.10.2.4 CR1 [1/2]

struct { ... } CR1

Flash control register 1 (FLASH_CR1)

6.10.2.5 CR1 [2/2]

struct { ... } CR1

Flash control register 1 (FLASH_CR1)

6.10.2.6 CR2 [1/2]

struct { ... } CR2

Flash control register 2 (FLASH_CR2)

6.10.2.7 CR2 [2/2]

struct { ... } CR2

Flash control register 2 (FLASH_CR2)

6.10.2.8 DUK

_BITS DUK

Data EEPROM write unlock key.

Definition at line 580 of file STM8AF_STM8S.h.

6.10.2.9 DUKR [1/2]

```
struct { ...  } DUKR
```

Data EEPROM unprotection key register (FLASH_DUKR)

6.10.2.10 DUKR [2/2]

```
struct { ...  } DUKR
```

Data EEPROM unprotection key register (FLASH_DUKR)

6.10.2.11 DUL

_BITS DUL

Data EEPROM area unlocked flag.

Definition at line 557 of file STM8AF_STM8S.h.

6.10.2.12 EOP

_BITS EOP

End of programming (write or erase operation) flag.

Definition at line 556 of file STM8AF_STM8S.h.

6.10.2.13 ERASE

_BITS ERASE

Block erasing.

Definition at line 521 of file STM8AF_STM8S.h.

6.10.2.14 FIX

_BITS FIX

Fixed Byte programming time.

Definition at line 508 of file STM8AF_STM8S.h.

6.10.2.15 FPR

struct { ... } FPR

Flash protection register (FLASH_FPR)

6.10.2.16 FPRG

_BITS FPRG

Fast block programming.

Definition at line 520 of file STM8AF_STM8S.h.

6.10.2.17 HALT

_BITS HALT

Power-down in Halt mode.

Definition at line 511 of file STM8AF_STM8S.h.

6.10.2.18 HVOFF

_BITS HVOFF

End of high voltage flag.

Definition at line 559 of file STM8AF_STM8S.h.

6.10.2.19 IAPSR [1/2]

```
struct { ...  } IAPSR
```

Flash status register (FLASH_IAPSR)

6.10.2.20 IAPSR [2/2]

```
struct { ...  } IAPSR
```

Flash status register (FLASH_IAPSR)

6.10.2.21 IE

```
_BITS IE
```

Flash Interrupt enable.

Definition at line 509 of file STM8AF_STM8S.h.

6.10.2.22 NCR2

```
struct { ...  } NCR2
```

Complementary flash control register 2 (FLASH_NCR2)

6.10.2.23 NERASE

```
_BITS NERASE
```

Block erasing.

Definition at line 532 of file STM8AF_STM8S.h.

6.10.2.24 NFPR

```
struct { ...  } NFPR
```

Complementary flash protection register (FLASH_NFPR)

6.10.2.25 NFPRG

_BITS NFPRG

Fast block programming.

Definition at line 531 of file STM8AF_STM8S.h.

6.10.2.26 NOPT

_BITS NOPT

Write option bytes.

Definition at line 534 of file STM8AF_STM8S.h.

6.10.2.27 NPRG

_BITS NPRG

Standard block programming.

Definition at line 529 of file STM8AF_STM8S.h.

6.10.2.28 NWPB

_BITS NWPB

User boot code area protection bits.

Definition at line 547 of file STM8AF_STM8S.h.

6.10.2.29 NWPRG

_BITS NWPRG

Word programming.

Definition at line 533 of file STM8AF_STM8S.h.

6.10.2.30 OPT

_BITS OPT

Write option bytes.

Definition at line 523 of file STM8AF_STM8S.h.

6.10.2.31 PRG

_BITS PRG

Standard block programming.

Definition at line 518 of file STM8AF_STM8S.h.

6.10.2.32 PUK

_BITS PUK

Program memory write unlock key.

Definition at line 570 of file STM8AF_STM8S.h.

6.10.2.33 PUKR [1/2]

struct { ... } PUKR

Flash program memory unprotecting key register (FLASH_PUKR)

6.10.2.34 PUKR [2/2]

struct { ... } PUKR

Flash program memory unprotecting key register (FLASH_PUKR)

6.10.2.35 PUL

_BITS PUL

Flash Program memory unlocked flag.

Definition at line 555 of file STM8AF_STM8S.h.

6.10.2.36 res

uint8_t res[2]

Reserved registers (2B)

Definition at line 565 of file STM8AF_STM8S.h.

6.10.2.37 res2

uint8_t res2[1]

Reserved register (1B)

Definition at line 575 of file STM8AF_STM8S.h.

6.10.2.38 WPB

_BITS WPB

User boot code area protection bits.

Definition at line 540 of file STM8AF_STM8S.h.

6.10.2.39 WPRG

_BITS WPRG

Word programming.

Definition at line 522 of file STM8AF_STM8S.h.

6.10.2.40 WR_PG_DIS

_BITS WR_PG_DIS

Write attempted to protected page flag.

Definition at line 554 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.11 I2C_t Struct Reference

struct for controlling I2C module (I2C)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS PE: 1
  Peripheral enable.
  _BITS __pad0__: 5
  _BITS ENGC: 1
  General call enable.
  _BITS NOSTRETCH: 1
  Clock stretching disable (Slave mode)
  } CR1

  I2C Control register 1 (I2C_CR1)
- struct {
  _BITS START: 1
  Start generation.
  _BITS STOP: 1
  Stop generation.
  _BITS ACK: 1
  Acknowledge enable.
  _BITS POS: 1
  Acknowledge position (for data reception)
  _BITS __pad0__: 3
  _BITS SWRST: 1
  Software reset.
  } CR2

  I2C Control register 2 (I2C_CR2)
- struct {
  _BITS FREQ: 6
  Peripheral clock frequency.
  _BITS __pad0__: 2
  } FREQR

I2C Frequency register (I2C_FREQR)

- struct {
  _BITS ADD0: 1
  Interface address [0] (in 10-bit address mode)
  _BITS ADD: 7
  Interface address [7:1].
  } OARL

I2C own address register low byte (I2C_OARL)

- struct {
  _BITS __pad0__: 1
  _BITS ADD: 2
  Interface address [9:8] (in 10-bit address mode)
  _BITS __pad1__: 3
  _BITS ADDCONF: 1
  Address mode configuration (must always be written as '1')
  _BITS ADDMODE: 1
  7-/10-bit addressing mode (Slave mode)
  } OARH

I2C own address register high byte (I2C_OARH)

- uint8_t res [1]

Reserved register (1B)

- struct {
  _BITS DATA: 8
  I2C data.
  } DR

I2C data register (I2C_DR)

- struct {
  _BITS SB: 1
  Start bit (Master mode)
  _BITS ADDR: 1
  Address sent (Master mode) / matched (slave mode)
  _BITS BTF: 1
  Byte transfer finished.
  _BITS ADD10: 1
  10-bit header sent (Master mode)
  _BITS STOPF: 1
  Stop detection (Slave mode)
  _BITS __pad0__: 1
  _BITS RXNE: 1
  Data register not empty (receivers)
  _BITS TXE: 1
  Data register empty (transmitters)
  } SR1

I2C Status register 1 (I2C_SR1)

- struct {
  _BITS BERR: 1
  Bus error.
  _BITS ARLO: 1
  Arbitration lost (Master mode)
  _BITS AF: 1
  Acknowledge failure.
  _BITS OVR: 1
  Overrun/underrun.

_BITS __pad0__: 1
_BITS WUFH: 1
Wakeup from Halt.
_BITS __pad1__: 2
} SR2

I2C Status register 2 (I2C_SR2)

- struct {
_BITS MSL: 1
Master/Slave.
_BITS BUSY: 1
Bus busy.
_BITS TRA: 1
Transmitter/Receiver.
_BITS __pad0__: 1
_BITS GENCALL: 1
General call header (Slavemode)
_BITS __pad1__: 3
} SR3

I2C Status register 3 (I2C_SR3)

- struct {
_BITS ITERREN: 1
Error interrupt enable.
_BITS ITEVTEN: 1
Event interrupt enable.
_BITS ITBUFEN: 1
Buffer interrupt enable.
_BITS __pad0__: 5
} ITR

I2C Interrupt register (I2C_ITR)

- struct {
_BITS CCR: 8
Clock control register (Master mode)
} CCRL

I2C Clock control register low byte (I2C_CCRL)

- struct {
_BITS CCR: 4
Clock control register in Fast/Standard mode (Master mode)
_BITS __pad0__: 2
_BITS DUTY: 1
Fast mode duty cycle.
_BITS FS: 1
I2C Master mode selection.
} CCRH

I2C Clock control register high byte (I2C_CCRH)

- struct {
_BITS TRISE: 6
Maximum rise time in Fast/Standard mode (Master mode)
_BITS __pad0__: 2
} TRISER

I2C rise time register (I2C_TRISER)

- uint8_t res2 [1]

Reserved register (1B). Was I2C packet error checking (undocumented in STM8 UM rev 9)

- struct {
  _BITS PE: 1
  Peripheral enable.
  _BITS __pad0__: 5
  _BITS ENGC: 1
  General call enable.
  _BITS NOSTRETCH: 1
  Clock stretching disable (Slave mode)
  } CR1

I2C Control register 1 (I2C_CR1)

- struct {
  _BITS START: 1
  Start generation.
  _BITS STOP: 1
  Stop generation.
  _BITS ACK: 1
  Acknowledge enable.
  _BITS POS: 1
  Acknowledge position (for data reception)
  _BITS __pad0__: 3
  _BITS SWRST: 1
  Software reset.
  } CR2

I2C Control register 2 (I2C_CR2)

- struct {
  _BITS FREQ: 6
  Peripheral clock frequency.
  _BITS __pad0__: 2
  } FREQR

I2C Frequency register (I2C_FREQR)

- struct {
  _BITS ADD0: 1
  Interface address [0] (in 10-bit address mode)
  _BITS ADD: 7
  Interface address [7:1].
  } OARL

I2C own address register low byte (I2C_OARL)

- struct {
  _BITS __pad0__: 1
  _BITS ADD: 2
  Interface address [9:8] (in 10-bit address mode)
  _BITS __pad1__: 3
  _BITS ADDCONF: 1
  Address mode configuration (must always be written as '1')
  _BITS ADDMODE: 1
  7-/10-bit addressing mode (Slave mode)
  } OARH

I2C own address register high byte (I2C_OARH)

- struct {
  _BITS DATA: 8
  I2C data.

} DR

I2C data register (I2C_DR)

- struct {
  _BITS SB: 1
  Start bit (Master mode)
  _BITS ADDR: 1
  Address sent (Master mode) / matched (Slave mode)
  _BITS BTF: 1
  Byte transfer finished.
  _BITS ADD10: 1
  10-bit header sent (Master mode)
  _BITS STOPF: 1
  Stop detection (Slave mode)
  _BITS __pad0__: 1
  _BITS RXNE: 1
  Data register not empty (receivers)
  _BITS TXE: 1
  Data register empty (transmitters)
  } SR1

I2C Status register 1 (I2C_SR1)

- struct {
  _BITS BERR: 1
  Bus error.
  _BITS ARLO: 1
  Arbitration lost (Master mode)
  _BITS AF: 1
  Acknowledge failure.
  _BITS OVR: 1
  Overrun/underrun.
  _BITS __pad0__: 1
  _BITS WUFH: 1
  Wakeup from Halt.
  _BITS __pad1__: 2
  } SR2

I2C Status register 2 (I2C_SR2)

- struct {
  _BITS MSL: 1
  Master/Slave.
  _BITS BUSY: 1
  Bus busy.
  _BITS TRA: 1
  Transmitter/Receiver.
  _BITS __pad0__: 1
  _BITS GENCALL: 1
  General call header (Slave mode)
  _BITS __pad1__: 2
  _BITS DUALF: 1
  Dual flag (Slave mode)
  } SR3

I2C Status register 3 (I2C_SR3)

- struct {
  _BITS ITERREN: 1
  Error interrupt enable.

_BITS ITEVTEN: 1
Event interrupt enable.
_BITS ITBUFEN: 1
Buffer interrupt enable.
_BITS __pad0__: 5
} ITR

I2C Interrupt register (I2C_ITR)

- struct {
_BITS CCR: 8
Clock control register (Master mode)
} CCRL

I2C Clock control register low byte (I2C_CCRL)

- struct {
_BITS CCR: 4
Clock control register in Fast/Standard mode (Master mode)
_BITS __pad0__: 2
_BITS DUTY: 1
Fast mode duty cycle.
_BITS FS: 1
I2C Master mode selection.
} CCRH

I2C Clock control register high byte (I2C_CCRH)

- struct {
_BITS TRISE: 6
Maximum rise time in Fast/Standard mode (Master mode)
_BITS __pad0__: 2
} TRISER

I2C rise time register (I2C_TRISER)

6.11.1 Detailed Description

struct for controlling I2C module (I2C)

Definition at line 1356 of file STM8AF_STM8S.h.

6.11.2 Field Documentation

6.11.2.1 __pad0__

```
_BITS __pad0__
```

Definition at line 1361 of file STM8AF_STM8S.h.

6.11.2.2 __pad1__

_BITS __pad1__

Definition at line 1396 of file STM8AF_STM8S.h.

6.11.2.3 ACK

_BITS ACK

Acknowledge enable.

Definition at line 1371 of file STM8AF_STM8S.h.

6.11.2.4 ADD

_BITS ADD

Interface address [7:1].

Interface address [9:8] (in 10-bit address mode)

Definition at line 1388 of file STM8AF_STM8S.h.

6.11.2.5 ADD0

_BITS ADD0

Interface address [0] (in 10-bit address mode)

Definition at line 1387 of file STM8AF_STM8S.h.

6.11.2.6 ADD10

_BITS ADD10

10-bit header sent (Master mode)

Definition at line 1417 of file STM8AF_STM8S.h.

6.11.2.7 ADDCONF

_BITS ADDCONF

Address mode configuration (must always be written as '1')

Definition at line 1397 of file STM8AF_STM8S.h.

6.11.2.8 ADDMODE

_BITS ADDMODE

7-/10-bit addressing mode (Slave mode)

Definition at line 1398 of file STM8AF_STM8S.h.

6.11.2.9 ADDR

_BITS ADDR

Address sent (Master mode) / matched (slave mode)

Address sent (Master mode) / matched (Slave mode)

Definition at line 1415 of file STM8AF_STM8S.h.

6.11.2.10 AF

_BITS AF

Acknowledge failure.

Definition at line 1429 of file STM8AF_STM8S.h.

6.11.2.11 ARLO

_BITS ARLO

Arbitration lost (Master mode)

Definition at line 1428 of file STM8AF_STM8S.h.

6.11.2.12 BERR

_BITS BERR

Bus error.

Definition at line 1427 of file STM8AF_STM8S.h.

6.11.2.13 BTF

_BITS BTF

Byte transfer finished.

Definition at line 1416 of file STM8AF_STM8S.h.

6.11.2.14 BUSY

_BITS BUSY

Bus busy.

Definition at line 1440 of file STM8AF_STM8S.h.

6.11.2.15 CCR

_BITS CCR

Clock control register (Master mode)

Clock control register in Fast/Standard mode (Master mode)

Definition at line 1459 of file STM8AF_STM8S.h.

6.11.2.16 CCRH [1/2]

struct { ... } CCRH

I2C Clock control register high byte (I2C_CCRH)

6.11.2.17 CCRH [2/2]

```
struct { ... } CCRH
```

I2C Clock control register high byte (I2C_CCRH)

6.11.2.18 CCRL [1/2]

```
struct { ... } CCRL
```

I2C Clock control register low byte (I2C_CCRL)

6.11.2.19 CCRL [2/2]

```
struct { ... } CCRL
```

I2C Clock control register low byte (I2C_CCRL)

6.11.2.20 CR1 [1/2]

```
struct { ... } CR1
```

I2C Control register 1 (I2C_CR1)

6.11.2.21 CR1 [2/2]

```
struct { ... } CR1
```

I2C Control register 1 (I2C_CR1)

6.11.2.22 CR2 [1/2]

```
struct { ... } CR2
```

I2C Control register 2 (I2C_CR2)

6.11.2.23 CR2 [2/2]

```
struct { ...  } CR2
```

I2C Control register 2 (I2C_CR2)

6.11.2.24 DATA

```
_BITS DATA
```

I2C data.

Definition at line 1408 of file STM8AF_STM8S.h.

6.11.2.25 DR [1/2]

```
struct { ...  } DR
```

I2C data register (I2C_DR)

6.11.2.26 DR [2/2]

```
struct { ...  } DR
```

I2C data register (I2C_DR)

6.11.2.27 DUALF

```
_BITS DUALF
```

Dual flag (Slave mode)

Definition at line 1156 of file STM8L10x.h.

6.11.2.28 DUTY

```
_BITS DUTY
```

Fast mode duty cycle.

Definition at line 1467 of file STM8AF_STM8S.h.

6.11.2.29 ENGC

_BITS ENGC

General call enable.

Definition at line 1362 of file STM8AF_STM8S.h.

6.11.2.30 FREQ

_BITS FREQ

Peripheral clock frequency.

Definition at line 1380 of file STM8AF_STM8S.h.

6.11.2.31 FREQR [1/2]

struct { ... } FREQR

I2C Frequency register (I2C_FREQR)

6.11.2.32 FREQR [2/2]

struct { ... } FREQR

I2C Frequency register (I2C_FREQR)

6.11.2.33 FS

_BITS FS

I2C Master mode selection.

Definition at line 1468 of file STM8AF_STM8S.h.

6.11.2.34 GENCALL

_BITS GENCALL

General call header (Slavemode)

General call header (Slave mode)

Definition at line 1443 of file STM8AF_STM8S.h.

6.11.2.35 ITBUFEN

_BITS ITBUFEN

Buffer interrupt enable.

Definition at line 1452 of file STM8AF_STM8S.h.

6.11.2.36 ITERREN

_BITS ITERREN

Error interrupt enable.

Definition at line 1450 of file STM8AF_STM8S.h.

6.11.2.37 ITEVTEN

_BITS ITEVTEN

Event interrupt enable.

Definition at line 1451 of file STM8AF_STM8S.h.

6.11.2.38 ITR [1/2]

struct { ... } ITR

I2C Interrupt register (I2C_ITR)

6.11.2.39 ITR [2/2]

```
struct { ...  } ITR
```

I2C Interrupt register (I2C_ITR)

6.11.2.40 MSL

```
_BITS MSL
```

Master/Slave.

Definition at line 1439 of file STM8AF_STM8S.h.

6.11.2.41 NOSTRETCH

```
_BITS NOSTRETCH
```

Clock stretching disable (Slave mode)

Definition at line 1363 of file STM8AF_STM8S.h.

6.11.2.42 OARH [1/2]

```
struct { ...  } OARH
```

I2C own address register high byte (I2C_OARH)

6.11.2.43 OARH [2/2]

```
struct { ...  } OARH
```

I2C own address register high byte (I2C_OARH)

6.11.2.44 OARL [1/2]

```
struct { ...  } OARL
```

I2C own address register low byte (I2C_OARL)

6.11.2.45 OARL [2/2]

```
struct { ...  } OARL
```

I2C own address register low byte (I2C_OARL)

6.11.2.46 OVR

```
_BITS OVR
```

Overrun/underrun.

Definition at line 1430 of file STM8AF_STM8S.h.

6.11.2.47 PE

```
_BITS PE
```

Peripheral enable.

Definition at line 1360 of file STM8AF_STM8S.h.

6.11.2.48 POS

```
_BITS POS
```

Acknowledge position (for data reception)

Definition at line 1372 of file STM8AF_STM8S.h.

6.11.2.49 res

```
uint8_t res
```

Reserved register (1B)

Definition at line 1403 of file STM8AF_STM8S.h.

6.11.2.50 res2

```
uint8_t res2[1]
```

Reserved register (1B). Was I2C packet error checking (undocumented in STM8 UM rev 9)

Definition at line 1480 of file STM8AF_STM8S.h.

6.11.2.51 RXNE

```
_BITS RXNE
```

Data register not empty (receivers)

Definition at line 1420 of file STM8AF_STM8S.h.

6.11.2.52 SB

```
_BITS SB
```

Start bit (Master mode)

Definition at line 1414 of file STM8AF_STM8S.h.

6.11.2.53 SR1 [1/2]

```
struct { ... } SR1
```

I2C Status register 1 (I2C_SR1)

6.11.2.54 SR1 [2/2]

```
struct { ... } SR1
```

I2C Status register 1 (I2C_SR1)

6.11.2.55 SR2 [1/2]

```
struct { ...  } SR2
```

I2C Status register 2 (I2C_SR2)

6.11.2.56 SR2 [2/2]

```
struct { ...  } SR2
```

I2C Status register 2 (I2C_SR2)

6.11.2.57 SR3 [1/2]

```
struct { ...  } SR3
```

I2C Status register 3 (I2C_SR3)

6.11.2.58 SR3 [2/2]

```
struct { ...  } SR3
```

I2C Status register 3 (I2C_SR3)

6.11.2.59 START

```
_BITS START
```

Start generation.

Definition at line 1369 of file STM8AF_STM8S.h.

6.11.2.60 STOP

```
_BITS STOP
```

Stop generation.

Definition at line 1370 of file STM8AF_STM8S.h.

6.11.2.61 STOPF

_BITS STOPF

Stop detection (Slave mode)

Definition at line 1418 of file STM8AF_STM8S.h.

6.11.2.62 SWRST

_BITS SWRST

Software reset.

Definition at line 1374 of file STM8AF_STM8S.h.

6.11.2.63 TRA

_BITS TRA

Transmitter/Receiver.

Definition at line 1441 of file STM8AF_STM8S.h.

6.11.2.64 TRISE

_BITS TRISE

Maximum rise time in Fast/Standard mode (Master mode)

Definition at line 1474 of file STM8AF_STM8S.h.

6.11.2.65 TRISER [1/2]

struct { ... } TRISER

I2C rise time register (I2C_TRISER)

6.11.2.66 TRISER [2/2]

```
struct { ...  } TRISER
```

I2C rise time register (I2C_TRISER)

6.11.2.67 TXE

```
_BITS TXE
```

Data register empty (transmitters)

Definition at line 1421 of file STM8AF_STM8S.h.

6.11.2.68 WUFH

```
_BITS WUFH
```

Wakeup from Halt.

Definition at line 1432 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.12 IRTIM_t Struct Reference

struct for Infrared Timer Module (IRTIM)

```
#include <STM8L10x.h>
```

Data Fields

- struct {
  _BITS IR_EN: 1
  Infrared output enable.
  _BITS HS_EN: 1
  High Sink LED driver enable.
  _BITS __pad0__: 6
  } CR

  IR-Timer Control register (IR_CR)

6.12.1 Detailed Description

struct for Infrared Timer Module (IRTIM)

Definition at line 2345 of file STM8L10x.h.

6.12.2 Field Documentation

6.12.2.1 __pad0__

_BITS __pad0__

Definition at line 2351 of file STM8L10x.h.

6.12.2.2 CR

struct { ... } CR

IR-Timer Control register (IR_CR)

6.12.2.3 HS_EN

_BITS HS_EN

High Sink LED driver enable.

Definition at line 2350 of file STM8L10x.h.

6.12.2.4 IR_EN

_BITS IR_EN

Infrared output enable.

Definition at line 2349 of file STM8L10x.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.13 ITC_t Struct Reference

struct for setting interrupt Priority (ITC)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS __pad0__: 2
  _BITS VECT1SPR: 2
  interrupt priority vector 1
  _BITS VECT2SPR: 2
  interrupt priority vector 2
  _BITS VECT3SPR: 2
  interrupt priority vector 3
  } SPR1

  interrupt priority register 1 (ITC_SPR1)
- struct {
  _BITS VECT4SPR: 2
  interrupt priority vector 4
  _BITS VECT5SPR: 2
  interrupt priority vector 5
  _BITS VECT6SPR: 2
  interrupt priority vector 6
  _BITS VECT7SPR: 2
  interrupt priority vector 7
  } SPR2

  interrupt priority register 2 (ITC_SPR2)
- struct {
  _BITS VECT8SPR: 2
  interrupt priority vector 8
  _BITS VECT9SPR: 2
  interrupt priority vector 9
  _BITS VECT10SPR: 2
  interrupt priority vector 10
  _BITS VECT11SPR: 2
  interrupt priority vector 11
  } SPR3

  interrupt priority register 3 (ITC_SPR3)
- struct {
  _BITS VECT12SPR: 2
  interrupt priority vector 12
  _BITS VECT13SPR: 2
  interrupt priority vector 13
  _BITS VECT14SPR: 2
  interrupt priority vector 14
  _BITS VECT15SPR: 2
  interrupt priority vector 15
  } SPR4

  interrupt priority register 4 (ITC_SPR4)

- struct {
  _BITS VECT16SPR: 2
  interrupt priority vector 16
  _BITS VECT17SPR: 2
  interrupt priority vector 17
  _BITS VECT18SPR: 2
  interrupt priority vector 18
  _BITS VECT19SPR: 2
  interrupt priority vector 19
  } SPR5

interrupt priority register 5 (ITC_SPR5)

- struct {
  _BITS VECT20SPR: 2
  interrupt priority vector 20
  _BITS VECT21SPR: 2
  interrupt priority vector 21
  _BITS VECT22SPR: 2
  interrupt priority vector 22
  _BITS VECT23SPR: 2
  interrupt priority vector 23
  } SPR6

interrupt priority register 6 (ITC_SPR6)

- struct {
  _BITS VECT24SPR: 2
  interrupt priority vector 24
  _BITS VECT25SPR: 2
  interrupt priority vector 25
  _BITS VECT26SPR: 2
  interrupt priority vector 26
  _BITS VECT27SPR: 2
  interrupt priority vector 27
  } SPR7

interrupt priority register 7 (ITC_SPR7)

- struct {
  _BITS VECT28SPR: 2
  interrupt priority vector 28
  _BITS VECT29SPR: 2
  interrupt priority vector 29
  _BITS __pad0__: 4
  } SPR8

interrupt priority register 8 (ITC_SPR8)

- struct {
  _BITS __pad0__: 2
  _BITS VECT1SPR: 2
  interrupt priority vector 1
  _BITS __pad1__: 4
  } SPR1

interrupt priority register 1 (ITC_SPR1)

- struct {
  _BITS VECT4SPR: 2
  interrupt priority vector 4
  _BITS __pad0__: 2

_BITS VECT6SPR: 2
interrupt priority vector 6
_BITS VECT7SPR: 2
interrupt priority vector 7
} SPR2

interrupt priority register 2 (ITC_SPR2)

- struct {
_BITS VECT8SPR: 2
interrupt priority vector 8
_BITS VECT9SPR: 2
interrupt priority vector 9
_BITS VECT10SPR: 2
interrupt priority vector 10
_BITS VECT11SPR: 2
interrupt priority vector 11
} SPR3

interrupt priority register 3 (ITC_SPR3)

- struct {
_BITS VECT12SPR: 2
interrupt priority vector 12
_BITS VECT13SPR: 2
interrupt priority vector 13
_BITS VECT14SPR: 2
interrupt priority vector 14
_BITS VECT15SPR: 2
interrupt priority vector 15
} SPR4

interrupt priority register 4 (ITC_SPR4)

- struct {
_BITS __pad0__: 6
_BITS VECT19SPR: 2
interrupt priority vector 19
} SPR5

interrupt priority register 5 (ITC_SPR5)

- struct {
_BITS VECT20SPR: 2
interrupt priority vector 20
_BITS VECT21SPR: 2
interrupt priority vector 21
_BITS VECT22SPR: 2
interrupt priority vector 22
_BITS __pad0__: 2
} SPR6

interrupt priority register 6 (ITC_SPR6)

- struct {
_BITS __pad0__: 2
_BITS VECT25SPR: 2
interrupt priority vector 25
_BITS VECT26SPR: 2
interrupt priority vector 26
_BITS VECT27SPR: 2
interrupt priority vector 27

} SPR7

interrupt priority register 7 (ITC_SPR7)

- struct {
  _BITS VECT28SPR: 2
  interrupt priority vector 28
  _BITS VECT29SPR: 2
  interrupt priority vector 29
  _BITS __pad0__: 4
  } SPR8

interrupt priority register 8 (ITC_SPR8)

6.13.1 Detailed Description

struct for setting interrupt Priority (ITC)

Definition at line 6326 of file STM8AF_STM8S.h.

6.13.2 Field Documentation

6.13.2.1 __pad0__

`_BITS __pad0__`

Definition at line 6330 of file STM8AF_STM8S.h.

6.13.2.2 __pad1__

`_BITS __pad1__`

Definition at line 2506 of file STM8L10x.h.

6.13.2.3 SPR1 [1/2]

`struct { ... } SPR1`

interrupt priority register 1 (ITC_SPR1)

6.13.2.4 SPR1 [2/2]

```
struct { ... } SPR1
```

interrupt priority register 1 (ITC_SPR1)

6.13.2.5 SPR2 [1/2]

```
struct { ... } SPR2
```

interrupt priority register 2 (ITC_SPR2)

6.13.2.6 SPR2 [2/2]

```
struct { ... } SPR2
```

interrupt priority register 2 (ITC_SPR2)

6.13.2.7 SPR3 [1/2]

```
struct { ... } SPR3
```

interrupt priority register 3 (ITC_SPR3)

6.13.2.8 SPR3 [2/2]

```
struct { ... } SPR3
```

interrupt priority register 3 (ITC_SPR3)

6.13.2.9 SPR4 [1/2]

```
struct { ... } SPR4
```

interrupt priority register 4 (ITC_SPR4)

6.13.2.10 SPR4 [2/2]

```
struct { ... } SPR4
```

interrupt priority register 4 (ITC_SPR4)

6.13.2.11 SPR5 [1/2]

```
struct { ... } SPR5
```

interrupt priority register 5 (ITC_SPR5)

6.13.2.12 SPR5 [2/2]

```
struct { ... } SPR5
```

interrupt priority register 5 (ITC_SPR5)

6.13.2.13 SPR6 [1/2]

```
struct { ... } SPR6
```

interrupt priority register 6 (ITC_SPR6)

6.13.2.14 SPR6 [2/2]

```
struct { ... } SPR6
```

interrupt priority register 6 (ITC_SPR6)

6.13.2.15 SPR7 [1/2]

```
struct { ... } SPR7
```

interrupt priority register 7 (ITC_SPR7)

6.13.2.16 SPR7 [2/2]

```
struct { ...  } SPR7
```

interrupt priority register 7 (ITC_SPR7)

6.13.2.17 SPR8 [1/2]

```
struct { ...  } SPR8
```

interrupt priority register 8 (ITC_SPR8)

6.13.2.18 SPR8 [2/2]

```
struct { ...  } SPR8
```

interrupt priority register 8 (ITC_SPR8)

6.13.2.19 VECT10SPR

```
_BITS VECT10SPR
```

interrupt priority vector 10

Definition at line 6350 of file STM8AF_STM8S.h.

6.13.2.20 VECT11SPR

```
_BITS VECT11SPR
```

interrupt priority vector 11

Definition at line 6351 of file STM8AF_STM8S.h.

6.13.2.21 VECT12SPR

```
_BITS VECT12SPR
```

interrupt priority vector 12

Definition at line 6357 of file STM8AF_STM8S.h.

6.13.2.22 VECT13SPR

_BITS VECT13SPR

interrupt priority vector 13

Definition at line 6358 of file STM8AF_STM8S.h.

6.13.2.23 VECT14SPR

_BITS VECT14SPR

interrupt priority vector 14

Definition at line 6359 of file STM8AF_STM8S.h.

6.13.2.24 VECT15SPR

_BITS VECT15SPR

interrupt priority vector 15

Definition at line 6360 of file STM8AF_STM8S.h.

6.13.2.25 VECT16SPR

_BITS VECT16SPR

interrupt priority vector 16

Definition at line 6366 of file STM8AF_STM8S.h.

6.13.2.26 VECT17SPR

_BITS VECT17SPR

interrupt priority vector 17

Definition at line 6367 of file STM8AF_STM8S.h.

6.13.2.27 VECT18SPR

_BITS VECT18SPR

interrupt priority vector 18

Definition at line 6368 of file STM8AF_STM8S.h.

6.13.2.28 VECT19SPR

_BITS VECT19SPR

interrupt priority vector 19

Definition at line 6369 of file STM8AF_STM8S.h.

6.13.2.29 VECT1SPR

_BITS VECT1SPR

interrupt priority vector 1

Definition at line 6331 of file STM8AF_STM8S.h.

6.13.2.30 VECT20SPR

_BITS VECT20SPR

interrupt priority vector 20

Definition at line 6375 of file STM8AF_STM8S.h.

6.13.2.31 VECT21SPR

_BITS VECT21SPR

interrupt priority vector 21

Definition at line 6376 of file STM8AF_STM8S.h.

6.13.2.32 VECT22SPR

_BITS VECT22SPR

interrupt priority vector 22

Definition at line 6377 of file STM8AF_STM8S.h.

6.13.2.33 VECT23SPR

_BITS VECT23SPR

interrupt priority vector 23

Definition at line 6378 of file STM8AF_STM8S.h.

6.13.2.34 VECT24SPR

_BITS VECT24SPR

interrupt priority vector 24

Definition at line 6384 of file STM8AF_STM8S.h.

6.13.2.35 VECT25SPR

_BITS VECT25SPR

interrupt priority vector 25

Definition at line 6385 of file STM8AF_STM8S.h.

6.13.2.36 VECT26SPR

_BITS VECT26SPR

interrupt priority vector 26

Definition at line 6386 of file STM8AF_STM8S.h.

6.13.2.37 VECT27SPR

_BITS VECT27SPR

interrupt priority vector 27

Definition at line 6387 of file STM8AF_STM8S.h.

6.13.2.38 VECT28SPR

_BITS VECT28SPR

interrupt priority vector 28

Definition at line 6393 of file STM8AF_STM8S.h.

6.13.2.39 VECT29SPR

_BITS VECT29SPR

interrupt priority vector 29

Definition at line 6394 of file STM8AF_STM8S.h.

6.13.2.40 VECT2SPR

_BITS VECT2SPR

interrupt priority vector 2

Definition at line 6332 of file STM8AF_STM8S.h.

6.13.2.41 VECT3SPR

_BITS VECT3SPR

interrupt priority vector 3

Definition at line 6333 of file STM8AF_STM8S.h.

6.13.2.42 VECT4SPR

_BITS VECT4SPR

interrupt priority vector 4

Definition at line 6339 of file STM8AF_STM8S.h.

6.13.2.43 VECT5SPR

_BITS VECT5SPR

interrupt priority vector 5

Definition at line 6340 of file STM8AF_STM8S.h.

6.13.2.44 VECT6SPR

_BITS VECT6SPR

interrupt priority vector 6

Definition at line 6341 of file STM8AF_STM8S.h.

6.13.2.45 VECT7SPR

_BITS VECT7SPR

interrupt priority vector 7

Definition at line 6342 of file STM8AF_STM8S.h.

6.13.2.46 VECT8SPR

_BITS VECT8SPR

interrupt priority vector 8

Definition at line 6348 of file STM8AF_STM8S.h.

6.13.2.47 VECT9SPR

_BITS VECT9SPR

interrupt priority vector 9

Definition at line 6349 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.14 IWDG_t Struct Reference

struct for access to Independent Timeout Watchdog registers (IWDG)

`#include <STM8AF_STM8S.h>`

Data Fields

- struct {
  _BITS KEY: 8
  IWDG Key.
  } KR

  IWDG Key register (IWDG_KR)
- struct {
  _BITS PRE: 3
  Prescaler divider.
  _BITS __pad0__: 5
  } PR

  IWDG Prescaler register (IWDG_PR)
- struct {
  _BITS RL: 8
  IWDG Reload value.
  } RLR

  IWDG Reload register (IWDG_RLR)
- struct {
  _BITS KEY: 8
  IWDG Key.
  } KR

  IWDG Key register (IWDG_KR)
- struct {
  _BITS PRE: 3
  Prescaler divider.
  _BITS __pad0__: 5
  } PR

  IWDG Prescaler register (IWDG_PR)
- struct {
  _BITS RL: 8
  IWDG Reload value.
  } RLR

  IWDG Reload register (IWDG_RLR)

6.14.1 Detailed Description

struct for access to Independent Timeout Watchdog registers (IWDG)

Definition at line 1046 of file STM8AF_STM8S.h.

6.14.2 Field Documentation

6.14.2.1 __pad0__

_BITS __pad0__

Definition at line 1057 of file STM8AF_STM8S.h.

6.14.2.2 KEY

_BITS KEY

IWDG Key.

Definition at line 1050 of file STM8AF_STM8S.h.

6.14.2.3 KR [1/2]

struct { ... } KR

IWDG Key register (IWDG_KR)

6.14.2.4 KR [2/2]

struct { ... } KR

IWDG Key register (IWDG_KR)

6.14.2.5 PR [1/2]

```
struct { ... } PR
```

IWDG Prescaler register (IWDG_PR)

6.14.2.6 PR [2/2]

```
struct { ... } PR
```

IWDG Prescaler register (IWDG_PR)

6.14.2.7 PRE

```
_BITS PRE
```

Prescaler divider.

Definition at line 1056 of file STM8AF_STM8S.h.

6.14.2.8 RL

```
_BITS RL
```

IWDG Reload value.

Definition at line 1063 of file STM8AF_STM8S.h.

6.14.2.9 RLR [1/2]

```
struct { ... } RLR
```

IWDG Reload register (IWDG_RLR)

6.14.2.10 RLR [2/2]

```
struct { ... } RLR
```

IWDG Reload register (IWDG_RLR)

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.15 PORT_t Struct Reference

structure for controlling pins in PORT mode (PORTx, x=A..I)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS PIN0: 1
  pin 0 output control
  _BITS PIN1: 1
  pin 1 output control
  _BITS PIN2: 1
  pin 2 output control
  _BITS PIN3: 1
  pin 3 output control
  _BITS PIN4: 1
  pin 4 output control
  _BITS PIN5: 1
  pin 5 output control
  _BITS PIN6: 1
  pin 6 output control
  _BITS PIN7: 1
  pin 7 output control
  } ODR

  Port x output data register (Px_ODR)
- struct {
  _BITS PIN0: 1
  pin 0 input control
  _BITS PIN1: 1
  pin 1 input control
  _BITS PIN2: 1
  pin 2 input control
  _BITS PIN3: 1
  pin 3 input control
  _BITS PIN4: 1
  pin 4 input control
  _BITS PIN5: 1
  pin 5 input control
  _BITS PIN6: 1

pin 6 input control
_BITS PIN7: 1
pin 7 input control
} IDR

Port x input data register (Px_IDR)

- struct {
_BITS PIN0: 1
pin 0 direction control
_BITS PIN1: 1
pin 1 direction control
_BITS PIN2: 1
pin 2 direction control
_BITS PIN3: 1
pin 3 direction control
_BITS PIN4: 1
pin 4 direction control
_BITS PIN5: 1
pin 5 direction control
_BITS PIN6: 1
pin 6 direction control
_BITS PIN7: 1
pin 7 direction control
} DDR

Port x data direction data register (Px_DDR)

- struct {
_BITS PIN0: 1
pin 0 control register 1
_BITS PIN1: 1
pin 1 control register 1
_BITS PIN2: 1
pin 2 control register 1
_BITS PIN3: 1
pin 3 control register 1
_BITS PIN4: 1
pin 4 control register 1
_BITS PIN5: 1
pin 5 control register 1
_BITS PIN6: 1
pin 6 control register 1
_BITS PIN7: 1
pin 7 control register 1
} CR1

Port x control register 1 (Px_CR1)

- struct {
_BITS PIN0: 1
pin 0 control register 2
_BITS PIN1: 1
pin 1 control register 2
_BITS PIN2: 1
pin 2 control register 2
_BITS PIN3: 1
pin 3 control register 2
_BITS PIN4: 1
pin 4 control register 2
_BITS PIN5: 1

pin 5 control register 2
_BITS PIN6: 1
pin 6 control register 2
_BITS PIN7: 1
pin 7 control register 2
} CR2

Port x control register 1 (Px_CR2)

- struct {
  _BITS PIN0: 1
  pin 0 output control
  _BITS PIN1: 1
  pin 1 output control
  _BITS PIN2: 1
  pin 2 output control
  _BITS PIN3: 1
  pin 3 output control
  _BITS PIN4: 1
  pin 4 output control
  _BITS PIN5: 1
  pin 5 output control
  _BITS PIN6: 1
  pin 6 output control
  _BITS PIN7: 1
  pin 7 output control
  } ODR

Port x output data register (Px_ODR)

- struct {
  _BITS PIN0: 1
  pin 0 input control
  _BITS PIN1: 1
  pin 1 input control
  _BITS PIN2: 1
  pin 2 input control
  _BITS PIN3: 1
  pin 3 input control
  _BITS PIN4: 1
  pin 4 input control
  _BITS PIN5: 1
  pin 5 input control
  _BITS PIN6: 1
  pin 6 input control
  _BITS PIN7: 1
  pin 7 input control
  } IDR

Port x input data register (Px_IDR)

- struct {
  _BITS PIN0: 1
  pin 0 direction control
  _BITS PIN1: 1
  pin 1 direction control
  _BITS PIN2: 1
  pin 2 direction control
  _BITS PIN3: 1
  pin 3 direction control
  _BITS PIN4: 1

pin 4 direction control
_BITS PIN5: 1
pin 5 direction control
_BITS PIN6: 1
pin 6 direction control
_BITS PIN7: 1
pin 7 direction control
} DDR

Port x data direction data register (Px_DDR)

- struct {
_BITS PIN0: 1
pin 0 control register 1
_BITS PIN1: 1
pin 1 control register 1
_BITS PIN2: 1
pin 2 control register 1
_BITS PIN3: 1
pin 3 control register 1
_BITS PIN4: 1
pin 4 control register 1
_BITS PIN5: 1
pin 5 control register 1
_BITS PIN6: 1
pin 6 control register 1
_BITS PIN7: 1
pin 7 control register 1
} CR1

Port x control register 1 (Px_CR1)

- struct {
_BITS PIN0: 1
pin 0 control register 2
_BITS PIN1: 1
pin 1 control register 2
_BITS PIN2: 1
pin 2 control register 2
_BITS PIN3: 1
pin 3 control register 2
_BITS PIN4: 1
pin 4 control register 2
_BITS PIN5: 1
pin 5 control register 2
_BITS PIN6: 1
pin 6 control register 2
_BITS PIN7: 1
pin 7 control register 2
} CR2

Port x control register 1 (Px_CR2)

6.15.1 Detailed Description

structure for controlling pins in PORT mode (PORTx, x=A..I)

Definition at line 324 of file STM8AF_STM8S.h.

6.15.2 Field Documentation

6.15.2.1 CR1 [1/2]

```
struct { ... } CR1
```

Port x control register 1 (Px_CR1)

6.15.2.2 CR1 [2/2]

```
struct { ... } CR1
```

Port x control register 1 (Px_CR1)

6.15.2.3 CR2 [1/2]

```
struct { ... } CR2
```

Port x control register 1 (Px_CR2)

6.15.2.4 CR2 [2/2]

```
struct { ... } CR2
```

Port x control register 1 (Px_CR2)

6.15.2.5 DDR [1/2]

```
struct { ... } DDR
```

Port x data direction data register (Px_DDR)

6.15.2.6 DDR [2/2]

```
struct { ...  } DDR
```

Port x data direction data register (Px_DDR)

6.15.2.7 IDR [1/2]

```
struct { ...  } IDR
```

Port x input data register (Px_IDR)

6.15.2.8 IDR [2/2]

```
struct { ...  } IDR
```

Port x input data register (Px_IDR)

6.15.2.9 ODR [1/2]

```
struct { ...  } ODR
```

Port x output data register (Px_ODR)

6.15.2.10 ODR [2/2]

```
struct { ...  } ODR
```

Port x output data register (Px_ODR)

6.15.2.11 PIN0

```
_BITS PIN0
```

pin 0 output control

pin 0 control register 2

pin 0 control register 1

pin 0 direction control

pin 0 input control

Definition at line 328 of file STM8AF_STM8S.h.

6.15.2.12 PIN1

_BITS PIN1

pin 1 output control

pin 1 control register 2

pin 1 control register 1

pin 1 direction control

pin 1 input control

Definition at line 329 of file STM8AF_STM8S.h.

6.15.2.13 PIN2

_BITS PIN2

pin 2 output control

pin 2 control register 2

pin 2 control register 1

pin 2 direction control

pin 2 input control

Definition at line 330 of file STM8AF_STM8S.h.

6.15.2.14 PIN3

_BITS PIN3

pin 3 output control

pin 3 control register 2

pin 3 control register 1

pin 3 direction control

pin 3 input control

Definition at line 331 of file STM8AF_STM8S.h.

6.15.2.15 PIN4

_BITS PIN4

pin 4 output control

pin 4 control register 2

pin 4 control register 1

pin 4 direction control

pin 4 input control

Definition at line 332 of file STM8AF_STM8S.h.

6.15.2.16 PIN5

_BITS PIN5

pin 5 output control

pin 5 control register 2

pin 5 control register 1

pin 5 direction control

pin 5 input control

Definition at line 333 of file STM8AF_STM8S.h.

6.15.2.17 PIN6

_BITS PIN6

pin 6 output control

pin 6 control register 2

pin 6 control register 1

pin 6 direction control

pin 6 input control

Definition at line 334 of file STM8AF_STM8S.h.

6.15.2.18 PIN7

_BITS PIN7

pin 7 output control

pin 7 control register 2

pin 7 control register 1

pin 7 direction control

pin 7 input control

Definition at line 335 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.16 RST_t Struct Reference

struct for determining reset source (RST)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS WWDGF: 1
  Window Watchdog reset flag.
  _BITS IWDGF: 1
  Independent Watchdog reset flag.
  _BITS ILLOPF: 1
  Illegal opcode reset flag.
  _BITS SWIMF: 1
  SWIM reset flag.
  _BITS EMCF: 1
  EMC reset flag.
  _BITS __pad0__: 3
  } SR

  Reset status register (RST_SR)
- struct {
  _BITS PIN_KEY: 8
  Reset pin configuration key.
  } CR

  Reset pin configuration register (RST_CR)

- struct {
  _BITS PORF: 1
  Power-on reset (POR) flag.
  _BITS IWDGF: 1
  Independent Watchdog reset flag.
  _BITS ILLOPF: 1
  Illegal opcode reset flag.
  _BITS SWIMF: 1
  SWIM reset flag.
  _BITS __pad0__: 4
  } SR

  Reset status register (RST_SR)

6.16.1 Detailed Description

struct for determining reset source (RST)

Definition at line 710 of file STM8AF_STM8S.h.

6.16.2 Field Documentation

6.16.2.1 __pad0__

```
_BITS __pad0__
```

Definition at line 719 of file STM8AF_STM8S.h.

6.16.2.2 CR

```
struct { ... } CR
```

Reset pin configuration register (RST_CR)

6.16.2.3 EMCF

```
_BITS EMCF
```

EMC reset flag.

Definition at line 718 of file STM8AF_STM8S.h.

6.16.2.4 ILLOPF

_BITS ILLOPF

Illegal opcode reset flag.

Definition at line 716 of file STM8AF_STM8S.h.

6.16.2.5 IWDGF

_BITS IWDGF

Independent Watchdog reset flag.

Definition at line 715 of file STM8AF_STM8S.h.

6.16.2.6 PIN_KEY

_BITS PIN_KEY

Reset pin configuration key.

Definition at line 654 of file STM8L10x.h.

6.16.2.7 PORF

_BITS PORF

Power-on reset (POR) flag.

Definition at line 659 of file STM8L10x.h.

6.16.2.8 SR [1/2]

struct { ... } SR

Reset status register (RST_SR)

6.16.2.9 SR [2/2]

```
struct { ...  } SR
```

Reset status register (RST_SR)

6.16.2.10 SWIMF

```
_BITS SWIMF
```

SWIM reset flag.

Definition at line 717 of file STM8AF_STM8S.h.

6.16.2.11 WWDGF

```
_BITS WWDGF
```

Window Watchdog reset flag.

Definition at line 714 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.17 SPI_t Struct Reference

struct for controlling SPI module (SPI)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS CPHA: 1
  Clock phase.
  _BITS CPOL: 1
  Clock polarity.
  _BITS MSTR: 1
  Master/slave selection.
  _BITS BR: 3
  Baudrate control.
  _BITS SPE: 1
  SPI enable.
  _BITS LSBFIRST: 1
  Frame format.
  } CR1

  SPI control register 1 (SPI_CR1)
- struct {
  _BITS SSI: 1
  Internal slave select.
  _BITS SSM: 1
  Software slave management.
  _BITS RXONLY: 1
  Receive only.
  _BITS __pad0__: 1
  _BITS CRCNEXT: 1
  Transmit CRC next.
  _BITS CRCEN: 1
  Hardware CRC calculation enable.
  _BITS BDOE: 1
  Input/Output enable in bidirectional mode.
  _BITS BDM: 1
  Bidirectional data mode enable.
  } CR2

  SPI control register 2 (SPI_CR2)
- struct {
  _BITS __pad0__: 4
  _BITS WKIE: 1
  Wakeup interrupt enable.
  _BITS ERRIE: 1
  Error interrupt enable.
  _BITS RXIE: 1
  Rx buffer not empty interrupt enable.
  _BITS TXIE: 1
  Tx buffer empty interrupt enable.
  } ICR

  SPI interrupt control register (SPI_ICR)
- struct {
  _BITS RXNE: 1
  Receive buffer not empty.
  _BITS TXE: 1
  Transmit buffer empty.
  _BITS __pad0__: 1
  _BITS WKUP: 1
  Wakeup flag.

_BITS CRCERR: 1
CRC error flag.
_BITS MODF: 1
Mode fault.
_BITS OVR: 1
Overrun flag.
_BITS BSY: 1
Busy flag.
} SR

SPI status register (SPI_SR)

- struct {
_BITS DATA: 8
SPI data.
} DR

SPI data register (SPI_DR)

- struct {
_BITS CRCPOLY: 8
CRC polynomial register.
} CRCPR

SPI CRC polynomial register (SPI_CRCPR)

- struct {
_BITS RxCRC: 8
Rx CRC Register.
} RXCRCR

SPI Rx CRC register (SPI_RXCRCR)

- struct {
_BITS TxCRC: 8
Tx CRC register.
} TXCRCR

SPI Tx CRC register (SPI_TXCRCR)

- struct {
_BITS CPHA: 1
Clock phase.
_BITS CPOL: 1
Clock polarity.
_BITS MSTR: 1
Master/slave selection.
_BITS BR: 3
Baudrate control.
_BITS SPE: 1
SPI enable.
_BITS LSBFIRST: 1
Frame format.
} CR1

SPI control register 1 (SPI_CR1)

- struct {
_BITS SSI: 1
Internal slave select.
_BITS SSM: 1
Software slave management.
_BITS RXONLY: 1

Receive only.
_BITS __pad0__: 3
_BITS BDOE: 1
Input/Output enable in bidirectional mode.
_BITS BDM: 1
Bidirectional data mode enable.
} CR2

SPI control register 2 (SPI_CR2)

- struct {
_BITS __pad0__: 4
_BITS WKIE: 1
Wakeup interrupt enable.
_BITS ERRIE: 1
Error interrupt enable.
_BITS RXIE: 1
Rx buffer not empty interrupt enable.
_BITS TXIE: 1
Tx buffer empty interrupt enable.
} ICR

SPI interrupt control register (SPI_ICR)

- struct {
_BITS RXNE: 1
Receive buffer not empty.
_BITS TXE: 1
Transmit buffer empty.
_BITS __pad0__: 1
_BITS WKUP: 1
Wakeup flag.
_BITS __pad1__: 1
_BITS MODF: 1
Mode fault.
_BITS OVR: 1
Overrun flag.
_BITS BSY: 1
Busy flag.
} SR

SPI status register (SPI_SR)

- struct {
_BITS DATA: 8
SPI data.
} DR

SPI data register (SPI_DR)

6.17.1 Detailed Description

struct for controlling SPI module (SPI)

Definition at line 1213 of file STM8AF_STM8S.h.

6.17.2 Field Documentation

6.17.2.1 __pad0__

_BITS __pad0__

Definition at line 1231 of file STM8AF_STM8S.h.

6.17.2.2 __pad1__

_BITS __pad1__

Definition at line 992 of file STM8L10x.h.

6.17.2.3 BDM

_BITS BDM

Bidirectional data mode enable.

Definition at line 1235 of file STM8AF_STM8S.h.

6.17.2.4 BDOE

_BITS BDOE

Input/Output enable in bidirectional mode.

Definition at line 1234 of file STM8AF_STM8S.h.

6.17.2.5 BR

_BITS BR

Baudrate control.

Definition at line 1220 of file STM8AF_STM8S.h.

6.17.2.6 BSY

_BITS BSY

Busy flag.

Definition at line 1258 of file STM8AF_STM8S.h.

6.17.2.7 CPHA

_BITS CPHA

Clock phase.

Definition at line 1217 of file STM8AF_STM8S.h.

6.17.2.8 CPOL

_BITS CPOL

Clock polarity.

Definition at line 1218 of file STM8AF_STM8S.h.

6.17.2.9 CR1 [1/2]

struct { ... } CR1

SPI control register 1 (SPI_CR1)

6.17.2.10 CR1 [2/2]

struct { ... } CR1

SPI control register 1 (SPI_CR1)

6.17.2.11 CR2 [1/2]

```
struct { ...  } CR2
```

SPI control register 2 (SPI_CR2)

6.17.2.12 CR2 [2/2]

```
struct { ...  } CR2
```

SPI control register 2 (SPI_CR2)

6.17.2.13 CRCEN

```
_BITS CRCEN
```

Hardware CRC calculation enable.

Definition at line 1233 of file STM8AF_STM8S.h.

6.17.2.14 CRCERR

```
_BITS CRCERR
```

CRC error flag.

Definition at line 1255 of file STM8AF_STM8S.h.

6.17.2.15 CRCNEXT

```
_BITS CRCNEXT
```

Transmit CRC next.

Definition at line 1232 of file STM8AF_STM8S.h.

6.17.2.16 CRCPOLY

_BITS CRCPOLY

CRC polynomial register.

Definition at line 1270 of file STM8AF_STM8S.h.

6.17.2.17 CRCPR

struct { ... } CRCPR

SPI CRC polynomial register (SPI_CRCPR)

6.17.2.18 DATA

_BITS DATA

SPI data.

Definition at line 1264 of file STM8AF_STM8S.h.

6.17.2.19 DR [1/2]

struct { ... } DR

SPI data register (SPI_DR)

6.17.2.20 DR [2/2]

struct { ... } DR

SPI data register (SPI_DR)

6.17.2.21 ERRIE

_BITS ERRIE

Error interrupt enable.

Definition at line 1243 of file STM8AF_STM8S.h.

6.17.2.22 ICR [1/2]

```
struct { ...  } ICR
```

SPI interrupt control register (SPI_ICR)

6.17.2.23 ICR [2/2]

```
struct { ...  } ICR
```

SPI interrupt control register (SPI_ICR)

6.17.2.24 LSBFIRST

```
_BITS LSBFIRST
```

Frame format.

Definition at line 1222 of file STM8AF_STM8S.h.

6.17.2.25 MODF

```
_BITS MODF
```

Mode fault.

Definition at line 1256 of file STM8AF_STM8S.h.

6.17.2.26 MSTR

```
_BITS MSTR
```

Master/slave selection.

Definition at line 1219 of file STM8AF_STM8S.h.

6.17.2.27 OVR

_BITS OVR

Overrun flag.

Definition at line 1257 of file STM8AF_STM8S.h.

6.17.2.28 RxCRC

_BITS RxCRC

Rx CRC Register.

Definition at line 1276 of file STM8AF_STM8S.h.

6.17.2.29 RXCRCR

struct { ... } RXCRCR

SPI Rx CRC register (SPI_RXCRCR)

6.17.2.30 RXIE

_BITS RXIE

Rx buffer not empty interrupt enable.

Definition at line 1244 of file STM8AF_STM8S.h.

6.17.2.31 RXNE

_BITS RXNE

Receive buffer not empty.

Definition at line 1251 of file STM8AF_STM8S.h.

6.17.2.32 RXONLY

_BITS RXONLY

Receive only.

Definition at line 1230 of file STM8AF_STM8S.h.

6.17.2.33 SPE

_BITS SPE

SPI enable.

Definition at line 1221 of file STM8AF_STM8S.h.

6.17.2.34 SR [1/2]

struct { ... } SR

SPI status register (SPI_SR)

6.17.2.35 SR [2/2]

struct { ... } SR

SPI status register (SPI_SR)

6.17.2.36 SSI

_BITS SSI

Internal slave select.

Definition at line 1228 of file STM8AF_STM8S.h.

6.17.2.37 SSM

_BITS SSM

Software slave management.

Definition at line 1229 of file STM8AF_STM8S.h.

6.17.2.38 TxCRC

_BITS TxCRC

Tx CRC register.

Definition at line 1282 of file STM8AF_STM8S.h.

6.17.2.39 TXCRCR

struct { ... } TXCRCR

SPI Tx CRC register (SPI_TXCRCR)

6.17.2.40 TXE

_BITS TXE

Transmit buffer empty.

Definition at line 1252 of file STM8AF_STM8S.h.

6.17.2.41 TXIE

_BITS TXIE

Tx buffer empty interrupt enable.

Definition at line 1245 of file STM8AF_STM8S.h.

6.17.2.42 WKIE

_BITS WKIE

Wakeup interrupt enable.

Definition at line 1242 of file STM8AF_STM8S.h.

6.17.2.43 WKUP

_BITS WKUP

Wakeup flag.

Definition at line 1254 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.18 TIM1_t Struct Reference

struct for controlling 16-Bit Timer 1 (TIM1)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS CEN: 1
  Counter enable.
  _BITS UDIS: 1
  Update disable.
  _BITS URS: 1
  Update request source.
  _BITS OPM: 1
  One-pulse mode.
  _BITS DIR: 1
  Direction.
  _BITS CMS: 2
  Center-aligned mode selection.
  _BITS ARPE: 1
  Auto-reload preload enable.
  } CR1

  TIM1 Control register 1 (TIM1_CR1)

- struct {
  _BITS CCPC: 1
  Capture/compare preloaded control.
  _BITS __pad0__: 1
  _BITS COMS: 1
  Capture/compare control update selection.
  _BITS __pad1__: 1
  _BITS MMS: 3
  Master mode selection.
  _BITS __pad2__: 1
  } CR2

TIM1 Control register 2 (TIM1_CR2)

- struct {
  _BITS SMS: 3
  Clock/trigger/slave mode selection.
  _BITS __pad0__: 1
  _BITS TS: 3
  Trigger selection.
  _BITS MSM: 1
  Master/slave mode.
  } SMCR

TIM1 Slave mode control register (TIM1_SMCR)

- struct {
  _BITS ETF: 4
  External trigger filter.
  _BITS ETPS: 2
  External trigger prescaler.
  _BITS ECE: 1
  External clock enable.
  _BITS ETP: 1
  External trigger polarity.
  } ETR

TIM1 External trigger register (TIM1_ETR)

- struct {
  _BITS UIE: 1
  Update interrupt enable.
  _BITS CC1IE: 1
  Capture/compare 1 interrupt enable.
  _BITS CC2IE: 1
  Capture/compare 2 interrupt enable.
  _BITS CC3IE: 1
  Capture/compare 3 interrupt enable.
  _BITS CC4IE: 1
  Capture/compare 4 interrupt enable.
  _BITS COMIE: 1
  Commutation interrupt enable.
  _BITS TIE: 1
  Trigger interrupt enable.
  _BITS BIE: 1
  Break interrupt enable.
  } IER

TIM1 Interrupt enable register (TIM1_IER)

- struct {

_BITS UIF: 1
Update interrupt flag.
_BITS CC1IF: 1
Capture/compare 1 interrupt flag.
_BITS CC2IF: 1
Capture/compare 2 interrupt flag.
_BITS CC3IF: 1
Capture/compare 3 interrupt flag.
_BITS CC4IF: 1
Capture/compare 4 interrupt flag.
_BITS COMIF: 1
Commutation interrupt flag.
_BITS TIF: 1
Trigger interrupt flag.
_BITS BIF: 1
Break interrupt flag.
} SR1

TIM1 Status register 1 (TIM1_SR1)

- struct {
_BITS __pad0__: 1
_BITS CC1OF: 1
Capture/compare 1 overcapture flag.
_BITS CC2OF: 1
Capture/compare 2 overcapture flag.
_BITS CC3OF: 1
Capture/compare 3 overcapture flag.
_BITS CC4OF: 1
Capture/compare 4 overcapture flag.
_BITS __pad1__: 3
} SR2

TIM1 Status register 2 (TIM1_SR2)

- struct {
_BITS UG: 1
Update generation.
_BITS CC1G: 1
Capture/compare 1 generation.
_BITS CC2G: 1
Capture/compare 2 generation.
_BITS CC3G: 1
Capture/compare 3 generation.
_BITS CC4G: 1
Capture/compare 4 generation.
_BITS COMG: 1
Capture/compare control update generation.
_BITS TG: 1
Trigger generation.
_BITS BG: 1
Break generation.
} EGR

TIM1 Event generation register (TIM1_EGR)

- union {
struct {
_BITS CC1S: 2
Compare 1 selection.
_BITS OC1FE: 1

Output compare 1 fast enable.
_BITS OC1PE: 1
Output compare 1 preload enable.
_BITS OC1M: 3
Output compare 1 mode.
_BITS OC1CE: 1
Output compare 1 clear enable.
} OUT
bitwise access to register (output mode)
struct {
_BITS CC1S: 2
Capture 1 selection.
_BITS IC1PSC: 2
Input capture 1 prescaler.
_BITS IC1F: 4
Input capture 1 filter.
} IN
bitwise access to register (input mode)
} CCMR1

TIM1 Capture/compare mode register 1 (TIM1_CCMR1)

- union {
struct {
_BITS CC2S: 2
Capture/compare 2 selection.
_BITS OC2FE: 1
Output compare 2 fast enable.
_BITS OC2PE: 1
Output compare 2 preload enable.
_BITS OC2M: 3
Output compare 2 mode.
_BITS OC2CE: 1
Output compare 2 clear enable.
} OUT
bitwise access to register (output mode)
struct {
_BITS CC2S: 2
Capture/compare 2 selection.
_BITS IC2PSC: 2
Input capture 2 prescaler.
_BITS IC2F: 4
Input capture 2 filter.
} IN
bitwise access to register (input mode)
} CCMR2

TIM1 Capture/compare mode register 2 (TIM1_CCMR2)

- union {
struct {
_BITS CC3S: 2
Capture/compare 3 selection.
_BITS OC3FE: 1
Output compare 3 fast enable.
_BITS OC3PE: 1
Output compare 3 preload enable.
_BITS OC3M: 3
Output compare 3 mode.
_BITS OC3CE: 1

Output compare 3 clear enable.

} OUT

bitwise access to register (output mode)

struct {

_BITS CC3S: 2

Capture/compare 3 selection.

_BITS IC3PSC: 2

Input capture 3 prescaler.

_BITS IC3F: 4

Input capture 3 filter.

} IN

bitwise access to register (input mode)

} CCMR3

TIM1 Capture/compare mode register 3 (TIM1_CCMR3)

- union {

struct {

_BITS CC4S: 2

Capture/compare 4 selection.

_BITS OC4FE: 1

Output compare 4 fast enable.

_BITS OC4PE: 1

Output compare 4 preload enable.

_BITS OC4M: 3

Output compare 4 mode.

_BITS OC4CE: 1

Output compare 4 clear enable.

} OUT

bitwise access to register (output mode)

struct {

_BITS CC4S: 2

Capture/compare 4 selection.

_BITS IC4PSC: 2

Input capture 4 prescaler.

_BITS IC4F: 4

Input capture 4 filter.

} IN

bitwise access to register (input mode)

} CCMR4

TIM1 Capture/compare mode register 4 (TIM1_CCMR4)

- struct {

_BITS CC1E: 1

Capture/compare 1 output enable.

_BITS CC1P: 1

Capture/compare 1 output polarity.

_BITS CC1NE: 1

Capture/compare 1 complementary output enable.

_BITS CC1NP: 1

Capture/compare 1 complementary output polarity.

_BITS CC2E: 1

Capture/compare 2 output enable.

_BITS CC2P: 1

Capture/compare 2 output polarity.

_BITS CC2NE: 1

Capture/compare 2 complementary output enable.

_BITS CC2NP: 1

Capture/compare 2 complementary output polarity.

} CCER1

TIM1 Capture/compare enable register 1 (TIM1_CCER1)

- struct {
  _BITS CC3E: 1
  Capture/compare 3 output enable.
  _BITS CC3P: 1
  Capture/compare 3 output polarity.
  _BITS CC3NE: 1
  Capture/compare 3 complementary output enable.
  _BITS CC3NP: 1
  Capture/compare 3 complementary output polarity.
  _BITS CC4E: 1
  Capture/compare 4 output enable.
  _BITS CC4P: 1
  Capture/compare 4 output polarity.
  _BITS __pad0__: 2
  } CCER2

TIM1 Capture/compare enable register 2 (TIM1_CCER2)

- struct {
  _BITS CNT: 8
  16-bit counter [15:8]
  } CNTRH

TIM1 16-bit counter high byte (TIM1_CNTRH)

- struct {
  _BITS CNT: 8
  16-bit counter [7:0]
  } CNTRL

TIM1 16-bit counter low byte (TIM1_CNTRL)

- struct {
  _BITS PSC: 8
  16-bit prescaler [15:8]
  } PSCRH

TIM1 16-bit prescaler high byte (TIM1_PSCRH)

- struct {
  _BITS PSC: 8
  16-bit prescaler [7:0]
  } PSCRL

TIM1 16-bit prescaler low byte (TIM1_PSCRL)

- struct {
  _BITS ARR: 8
  16-bit auto-reload value [15:8]
  } ARRH

TIM1 16-bit auto-reload value high byte (TIM1_ARRH)

- struct {
  _BITS ARR: 8
  16-bit auto-reload value [7:0]
  } ARRL

TIM1 16-bit auto-reload value low byte (TIM1_ARRL)

- struct {
  _BITS REP: 8
  Repetition counter value.
  } RCR

TIM1 Repetition counter (TIM1_RCR)

- struct {
  _BITS CCR1: 8
  16-bit capture/compare value 1 [15:8]
  } CCR1H

TIM1 16-bit capture/compare value 1 high byte (TIM1_CCR1H)

- struct {
  _BITS CCR1: 8
  16-bit capture/compare value 1 [7:0]
  } CCR1L

TIM1 16-bit capture/compare value 1 low byte (TIM1_CCR1L)

- struct {
  _BITS CCR2: 8
  16-bit capture/compare value 2 [15:8]
  } CCR2H

TIM1 16-bit capture/compare value 2 high byte (TIM1_CCR2H)

- struct {
  _BITS CCR2: 8
  16-bit capture/compare value 2 [7:0]
  } CCR2L

TIM1 16-bit capture/compare value 2 low byte (TIM1_CCR2L)

- struct {
  _BITS CCR3: 8
  16-bit capture/compare value 3 [15:8]
  } CCR3H

TIM1 16-bit capture/compare value 3 high byte (TIM1_CCR3H)

- struct {
  _BITS CCR3: 8
  16-bit capture/compare value 3 [7:0]
  } CCR3L

TIM1 16-bit capture/compare value 3 low byte (TIM1_CCR3L)

- struct {
  _BITS CCR4: 8
  16-bit capture/compare value 4 [15:8]
  } CCR4H

TIM1 16-bit capture/compare value 4 high byte (TIM1_CCR4H)

- struct {
  _BITS CCR4: 8
  16-bit capture/compare value 4 [7:0]
  } CCR4L

TIM1 16-bit capture/compare value 4 low byte (TIM1_CCR4L)

- struct {
  _BITS LOCK: 2
  Lock configuration.
  _BITS OSSI: 1
  Off state selection for idle mode.
  _BITS OSSR: 1
  Off state selection for Run mode.
  _BITS BKE: 1
  Break enable.
  _BITS BKP: 1
  Break polarity.
  _BITS AOE: 1
  Automatic output enable.
  _BITS MOE: 1
  Main output enable.
  } BKR

  TIM1 Break register (TIM1_BKR)
- struct {
  _BITS DTG: 8
  Deadtime generator set-up.
  } DTR

  TIM1 Dead-time register (TIM1_DTR)
- struct {
  _BITS OIS1: 1
  Output idle state 1 (OC1 output)
  _BITS OIS1N: 1
  Output idle state 1 (OC1N output)
  _BITS OIS2: 1
  Output idle state 2 (OC2 output)
  _BITS OIS2N: 1
  Output idle state 2 (OC2N output)
  _BITS OIS3: 1
  Output idle state 3 (OC3 output)
  _BITS OIS3N: 1
  Output idle state 3 (OC3N output)
  _BITS OIS4: 1
  Output idle state 4 (OC4 output)
  _BITS __pad0__: 1
  } OISR

  TIM1 Output idle state register (TIM1_OISR)

6.18.1 Detailed Description

struct for controlling 16-Bit Timer 1 (TIM1)

Definition at line 2474 of file STM8AF_STM8S.h.

6.18.2 Field Documentation

6.18.2.1 __pad0__

_BITS __pad0__

Definition at line 2491 of file STM8AF_STM8S.h.

6.18.2.2 __pad1__

_BITS __pad1__

Definition at line 2493 of file STM8AF_STM8S.h.

6.18.2.3 __pad2__

_BITS __pad2__

Definition at line 2495 of file STM8AF_STM8S.h.

6.18.2.4 AOE

_BITS AOE

Automatic output enable.

Definition at line 2777 of file STM8AF_STM8S.h.

6.18.2.5 ARPE

_BITS ARPE

Auto-reload preload enable.

Definition at line 2484 of file STM8AF_STM8S.h.

6.18.2.6 ARR

_BITS ARR

16-bit auto-reload value [15:8]

16-bit auto-reload value [7:0]

Definition at line 2706 of file STM8AF_STM8S.h.

6.18.2.7 ARRH

struct { ... } ARRH

TIM1 16-bit auto-reload value high byte (TIM1_ARRH)

6.18.2.8 ARRL

struct { ... } ARRL

TIM1 16-bit auto-reload value low byte (TIM1_ARRL)

6.18.2.9 BG

_BITS BG

Break generation.

Definition at line 2563 of file STM8AF_STM8S.h.

6.18.2.10 BIE

_BITS BIE

Break interrupt enable.

Definition at line 2526 of file STM8AF_STM8S.h.

6.18.2.11 BIF

_BITS BIF

Break interrupt flag.

Definition at line 2539 of file STM8AF_STM8S.h.

6.18.2.12 BKE

_BITS BKE

Break enable.

Definition at line 2775 of file STM8AF_STM8S.h.

6.18.2.13 BKP

_BITS BKP

Break polarity.

Definition at line 2776 of file STM8AF_STM8S.h.

6.18.2.14 BKR

struct { ... } BKR

TIM1 Break register (TIM1_BKR)

6.18.2.15 CC1E

_BITS CC1E

Capture/compare 1 output enable.

Definition at line 2657 of file STM8AF_STM8S.h.

6.18.2.16 CC1G

_BITS CC1G

Capture/compare 1 generation.

Definition at line 2557 of file STM8AF_STM8S.h.

6.18.2.17 CC1IE

_BITS CC1IE

Capture/compare 1 interrupt enable.

Definition at line 2520 of file STM8AF_STM8S.h.

6.18.2.18 CC1IF

_BITS CC1IF

Capture/compare 1 interrupt flag.

Definition at line 2533 of file STM8AF_STM8S.h.

6.18.2.19 CC1NE

_BITS CC1NE

Capture/compare 1 complementary output enable.

Definition at line 2659 of file STM8AF_STM8S.h.

6.18.2.20 CC1NP

_BITS CC1NP

Capture/compare 1 complementary output polarity.

Definition at line 2660 of file STM8AF_STM8S.h.

6.18.2.21 CC1OF

_BITS CC1OF

Capture/compare 1 overcapture flag.

Definition at line 2546 of file STM8AF_STM8S.h.

6.18.2.22 CC1P

_BITS CC1P

Capture/compare 1 output polarity.

Definition at line 2658 of file STM8AF_STM8S.h.

6.18.2.23 CC1S

_BITS CC1S

Compare 1 selection.

Capture 1 selection.

Definition at line 2572 of file STM8AF_STM8S.h.

6.18.2.24 CC2E

_BITS CC2E

Capture/compare 2 output enable.

Definition at line 2661 of file STM8AF_STM8S.h.

6.18.2.25 CC2G

_BITS CC2G

Capture/compare 2 generation.

Definition at line 2558 of file STM8AF_STM8S.h.

6.18.2.26 CC2IE

_BITS CC2IE

Capture/compare 2 interrupt enable.

Definition at line 2521 of file STM8AF_STM8S.h.

6.18.2.27 CC2IF

_BITS CC2IF

Capture/compare 2 interrupt flag.

Definition at line 2534 of file STM8AF_STM8S.h.

6.18.2.28 CC2NE

_BITS CC2NE

Capture/compare 2 complementary output enable.

Definition at line 2663 of file STM8AF_STM8S.h.

6.18.2.29 CC2NP

_BITS CC2NP

Capture/compare 2 complementary output polarity.

Definition at line 2664 of file STM8AF_STM8S.h.

6.18.2.30 CC2OF

_BITS CC2OF

Capture/compare 2 overcapture flag.

Definition at line 2547 of file STM8AF_STM8S.h.

6.18.2.31 CC2P

_BITS CC2P

Capture/compare 2 output polarity.

Definition at line 2662 of file STM8AF_STM8S.h.

6.18.2.32 CC2S

_BITS CC2S

Capture/compare 2 selection.

Definition at line 2594 of file STM8AF_STM8S.h.

6.18.2.33 CC3E

_BITS CC3E

Capture/compare 3 output enable.

Definition at line 2670 of file STM8AF_STM8S.h.

6.18.2.34 CC3G

_BITS CC3G

Capture/compare 3 generation.

Definition at line 2559 of file STM8AF_STM8S.h.

6.18.2.35 CC3IE

_BITS CC3IE

Capture/compare 3 interrupt enable.

Definition at line 2522 of file STM8AF_STM8S.h.

6.18.2.36 CC3IF

_BITS CC3IF

Capture/compare 3 interrupt flag.

Definition at line 2535 of file STM8AF_STM8S.h.

6.18.2.37 CC3NE

_BITS CC3NE

Capture/compare 3 complementary output enable.

Definition at line 2672 of file STM8AF_STM8S.h.

6.18.2.38 CC3NP

_BITS CC3NP

Capture/compare 3 complementary output polarity.

Definition at line 2673 of file STM8AF_STM8S.h.

6.18.2.39 CC3OF

_BITS CC3OF

Capture/compare 3 overcapture flag.

Definition at line 2548 of file STM8AF_STM8S.h.

6.18.2.40 CC3P

_BITS CC3P

Capture/compare 3 output polarity.

Definition at line 2671 of file STM8AF_STM8S.h.

6.18.2.41 CC3S

_BITS CC3S

Capture/compare 3 selection.

Definition at line 2616 of file STM8AF_STM8S.h.

6.18.2.42 CC4E

_BITS CC4E

Capture/compare 4 output enable.

Definition at line 2674 of file STM8AF_STM8S.h.

6.18.2.43 CC4G

_BITS CC4G

Capture/compare 4 generation.

Definition at line 2560 of file STM8AF_STM8S.h.

6.18.2.44 CC4IE

_BITS CC4IE

Capture/compare 4 interrupt enable.

Definition at line 2523 of file STM8AF_STM8S.h.

6.18.2.45 CC4IF

_BITS CC4IF

Capture/compare 4 interrupt flag.

Definition at line 2536 of file STM8AF_STM8S.h.

6.18.2.46 CC4OF

_BITS CC4OF

Capture/compare 4 overcapture flag.

Definition at line 2549 of file STM8AF_STM8S.h.

6.18.2.47 CC4P

_BITS CC4P

Capture/compare 4 output polarity.

Definition at line 2675 of file STM8AF_STM8S.h.

6.18.2.48 CC4S

_BITS CC4S

Capture/compare 4 selection.

Definition at line 2638 of file STM8AF_STM8S.h.

6.18.2.49 CCER1

struct { ... } CCER1

TIM1 Capture/compare enable register 1 (TIM1_CCER1)

6.18.2.50 CCER2

struct { ... } CCER2

TIM1 Capture/compare enable register 2 (TIM1_CCER2)

6.18.2.51 CCMR1

```
union { ... } CCMR1
```

TIM1 Capture/compare mode register 1 (TIM1_CCMR1)

6.18.2.52 CCMR2

```
union { ... } CCMR2
```

TIM1 Capture/compare mode register 2 (TIM1_CCMR2)

6.18.2.53 CCMR3

```
union { ... } CCMR3
```

TIM1 Capture/compare mode register 3 (TIM1_CCMR3)

6.18.2.54 CCMR4

```
union { ... } CCMR4
```

TIM1 Capture/compare mode register 4 (TIM1_CCMR4)

6.18.2.55 CCPC

```
_BITS CCPC
```

Capture/compare preloaded control.

Definition at line 2490 of file STM8AF_STM8S.h.

6.18.2.56 CCR1

```
_BITS CCR1
```

16-bit capture/compare value 1 [15:8]

16-bit capture/compare value 1 [7:0]

Definition at line 2724 of file STM8AF_STM8S.h.

6.18.2.57 CCR1H

```
struct { ...  } CCR1H
```

TIM1 16-bit capture/compare value 1 high byte (TIM1_CCR1H)

6.18.2.58 CCR1L

```
struct { ...  } CCR1L
```

TIM1 16-bit capture/compare value 1 low byte (TIM1_CCR1L)

6.18.2.59 CCR2

```
_BITS CCR2
```

16-bit capture/compare value 2 [15:8]

16-bit capture/compare value 2 [7:0]

Definition at line 2736 of file STM8AF_STM8S.h.

6.18.2.60 CCR2H

```
struct { ...  } CCR2H
```

TIM1 16-bit capture/compare value 2 high byte (TIM1_CCR2H)

6.18.2.61 CCR2L

```
struct { ...  } CCR2L
```

TIM1 16-bit capture/compare value 2 low byte (TIM1_CCR2L)

6.18.2.62 CCR3

```
_BITS CCR3
```

16-bit capture/compare value 3 [15:8]

16-bit capture/compare value 3 [7:0]

Definition at line 2748 of file STM8AF_STM8S.h.

6.18.2.63 CCR3H

```
struct { ...  } CCR3H
```

TIM1 16-bit capture/compare value 3 high byte (TIM1_CCR3H)

6.18.2.64 CCR3L

```
struct { ...  } CCR3L
```

TIM1 16-bit capture/compare value 3 low byte (TIM1_CCR3L)

6.18.2.65 CCR4

```
_BITS CCR4
```

16-bit capture/compare value 4 [15:8]

16-bit capture/compare value 4 [7:0]

Definition at line 2760 of file STM8AF_STM8S.h.

6.18.2.66 CCR4H

```
struct { ...  } CCR4H
```

TIM1 16-bit capture/compare value 4 high byte (TIM1_CCR4H)

6.18.2.67 CCR4L

```
struct { ...  } CCR4L
```

TIM1 16-bit capture/compare value 4 low byte (TIM1_CCR4L)

6.18.2.68 CEN

```
_BITS CEN
```

Counter enable.

Definition at line 2478 of file STM8AF_STM8S.h.

6.18.2.69 CMS

```
_BITS CMS
```

Center-aligned mode selection.

Definition at line 2483 of file STM8AF_STM8S.h.

6.18.2.70 CNT

```
_BITS CNT
```

16-bit counter [15:8]

16-bit counter [7:0]

Definition at line 2682 of file STM8AF_STM8S.h.

6.18.2.71 CNTRH

```
struct { ...  } CNTRH
```

TIM1 16-bit counter high byte (TIM1_CNTRH)

6.18.2.72 CNTRL

```
struct { ...  } CNTRL
```

TIM1 16-bit counter low byte (TIM1_CNTRL)

6.18.2.73 COMG

```
_BITS COMG
```

Capture/compare control update generation.

Definition at line 2561 of file STM8AF_STM8S.h.

6.18.2.74 COMIE

```
_BITS COMIE
```

Commutation interrupt enable.

Definition at line 2524 of file STM8AF_STM8S.h.

6.18.2.75 COMIF

```
_BITS COMIF
```

Commutation interrupt flag.

Definition at line 2537 of file STM8AF_STM8S.h.

6.18.2.76 COMS

```
_BITS COMS
```

Capture/compare control update selection.

Definition at line 2492 of file STM8AF_STM8S.h.

6.18.2.77 CR1

```
struct { ...  } CR1
```

TIM1 Control register 1 (TIM1_CR1)

6.18.2.78 CR2

```
struct { ...  } CR2
```

TIM1 Control register 2 (TIM1_CR2)

6.18.2.79 DIR

```
_BITS DIR
```

Direction.

Definition at line 2482 of file STM8AF_STM8S.h.

6.18.2.80 DTG

```
_BITS DTG
```

Deadtime generator set-up.

Definition at line 2784 of file STM8AF_STM8S.h.

6.18.2.81 DTR

```
struct { ...  } DTR
```

TIM1 Dead-time register (TIM1_DTR)

6.18.2.82 ECE

```
_BITS ECE
```

External clock enable.

Definition at line 2512 of file STM8AF_STM8S.h.

6.18.2.83 EGR

```
struct { ...  } EGR
```

TIM1 Event generation register (TIM1_EGR)

6.18.2.84 ETF

```
_BITS ETF
```

External trigger filter.

Definition at line 2510 of file STM8AF_STM8S.h.

6.18.2.85 ETP

```
_BITS ETP
```

External trigger polarity.

Definition at line 2513 of file STM8AF_STM8S.h.

6.18.2.86 ETPS

```
_BITS ETPS
```

External trigger prescaler.

Definition at line 2511 of file STM8AF_STM8S.h.

6.18.2.87 ETR

```
struct { ...  } ETR
```

TIM1 External trigger register (TIM1_ETR)

6.18.2.88 IC1F

_BITS IC1F

Input capture 1 filter.

Definition at line 2583 of file STM8AF_STM8S.h.

6.18.2.89 IC1PSC

_BITS IC1PSC

Input capture 1 prescaler.

Definition at line 2582 of file STM8AF_STM8S.h.

6.18.2.90 IC2F

_BITS IC2F

Input capture 2 filter.

Definition at line 2605 of file STM8AF_STM8S.h.

6.18.2.91 IC2PSC

_BITS IC2PSC

Input capture 2 prescaler.

Definition at line 2604 of file STM8AF_STM8S.h.

6.18.2.92 IC3F

_BITS IC3F

Input capture 3 filter.

Definition at line 2627 of file STM8AF_STM8S.h.

6.18.2.93 IC3PSC

_BITS IC3PSC

Input capture 3 prescaler.

Definition at line 2626 of file STM8AF_STM8S.h.

6.18.2.94 IC4F

_BITS IC4F

Input capture 4 filter.

Definition at line 2649 of file STM8AF_STM8S.h.

6.18.2.95 IC4PSC

_BITS IC4PSC

Input capture 4 prescaler.

Definition at line 2648 of file STM8AF_STM8S.h.

6.18.2.96 IER

struct { ... } IER

TIM1 Interrupt enable register (TIM1_IER)

6.18.2.97 IN [1/4]

struct { ... } IN

bitwise access to register (input mode)

6.18.2.98 IN [2/4]

```
struct { ...  } IN
```

bitwise access to register (input mode)

6.18.2.99 IN [3/4]

```
struct { ...  } IN
```

bitwise access to register (input mode)

6.18.2.100 IN [4/4]

```
struct { ...  } IN
```

bitwise access to register (input mode)

6.18.2.101 LOCK

```
_BITS LOCK
```

Lock configuration.

Definition at line 2772 of file STM8AF_STM8S.h.

6.18.2.102 MMS

```
_BITS MMS
```

Master mode selection.

Definition at line 2494 of file STM8AF_STM8S.h.

6.18.2.103 MOE

```
_BITS MOE
```

Main output enable.

Definition at line 2778 of file STM8AF_STM8S.h.

6.18.2.104 MSM

_BITS MSM

Master/slave mode.

Definition at line 2504 of file STM8AF_STM8S.h.

6.18.2.105 OC1CE

_BITS OC1CE

Output compare 1 clear enable.

Definition at line 2576 of file STM8AF_STM8S.h.

6.18.2.106 OC1FE

_BITS OC1FE

Output compare 1 fast enable.

Definition at line 2573 of file STM8AF_STM8S.h.

6.18.2.107 OC1M

_BITS OC1M

Output compare 1 mode.

Definition at line 2575 of file STM8AF_STM8S.h.

6.18.2.108 OC1PE

_BITS OC1PE

Output compare 1 preload enable.

Definition at line 2574 of file STM8AF_STM8S.h.

6.18.2.109 OC2CE

_BITS OC2CE

Output compare 2 clear enable.

Definition at line 2598 of file STM8AF_STM8S.h.

6.18.2.110 OC2FE

_BITS OC2FE

Output compare 2 fast enable.

Definition at line 2595 of file STM8AF_STM8S.h.

6.18.2.111 OC2M

_BITS OC2M

Output compare 2 mode.

Definition at line 2597 of file STM8AF_STM8S.h.

6.18.2.112 OC2PE

_BITS OC2PE

Output compare 2 preload enable.

Definition at line 2596 of file STM8AF_STM8S.h.

6.18.2.113 OC3CE

_BITS OC3CE

Output compare 3 clear enable.

Definition at line 2620 of file STM8AF_STM8S.h.

6.18.2.114 OC3FE

_BITS OC3FE

Output compare 3 fast enable.

Definition at line 2617 of file STM8AF_STM8S.h.

6.18.2.115 OC3M

_BITS OC3M

Output compare 3 mode.

Definition at line 2619 of file STM8AF_STM8S.h.

6.18.2.116 OC3PE

_BITS OC3PE

Output compare 3 preload enable.

Definition at line 2618 of file STM8AF_STM8S.h.

6.18.2.117 OC4CE

_BITS OC4CE

Output compare 4 clear enable.

Definition at line 2642 of file STM8AF_STM8S.h.

6.18.2.118 OC4FE

_BITS OC4FE

Output compare 4 fast enable.

Definition at line 2639 of file STM8AF_STM8S.h.

6.18.2.119 OC4M

_BITS OC4M

Output compare 4 mode.

Definition at line 2641 of file STM8AF_STM8S.h.

6.18.2.120 OC4PE

_BITS OC4PE

Output compare 4 preload enable.

Definition at line 2640 of file STM8AF_STM8S.h.

6.18.2.121 OIS1

_BITS OIS1

Output idle state 1 (OC1 output)

Definition at line 2790 of file STM8AF_STM8S.h.

6.18.2.122 OIS1N

_BITS OIS1N

Output idle state 1 (OC1N output)

Definition at line 2791 of file STM8AF_STM8S.h.

6.18.2.123 OIS2

_BITS OIS2

Output idle state 2 (OC2 output)

Definition at line 2792 of file STM8AF_STM8S.h.

6.18.2.124 OIS2N

```
_BITS OIS2N
```

Output idle state 2 (OC2N output)

Definition at line 2793 of file STM8AF_STM8S.h.

6.18.2.125 OIS3

```
_BITS OIS3
```

Output idle state 3 (OC3 output)

Definition at line 2794 of file STM8AF_STM8S.h.

6.18.2.126 OIS3N

```
_BITS OIS3N
```

Output idle state 3 (OC3N output)

Definition at line 2795 of file STM8AF_STM8S.h.

6.18.2.127 OIS4

```
_BITS OIS4
```

Output idle state 4 (OC4 output)

Definition at line 2796 of file STM8AF_STM8S.h.

6.18.2.128 OISR

```
struct { ...  } OISR
```

TIM1 Output idle state register (TIM1_OISR)

6.18.2.129 OPM

_BITS OPM

One-pulse mode.

Definition at line 2481 of file STM8AF_STM8S.h.

6.18.2.130 OSSI

_BITS OSSI

Off state selection for idle mode.

Definition at line 2773 of file STM8AF_STM8S.h.

6.18.2.131 OSSR

_BITS OSSR

Off state selection for Run mode.

Definition at line 2774 of file STM8AF_STM8S.h.

6.18.2.132 OUT [1/4]

struct { ... } OUT

bitwise access to register (output mode)

6.18.2.133 OUT [2/4]

struct { ... } OUT

bitwise access to register (output mode)

6.18.2.134 OUT [3/4]

```
struct { ...  } OUT
```

bitwise access to register (output mode)

6.18.2.135 OUT [4/4]

```
struct { ...  } OUT
```

bitwise access to register (output mode)

6.18.2.136 PSC

```
_BITS PSC
```

16-bit prescaler [15:8]

16-bit prescaler [7:0]

Definition at line 2694 of file STM8AF_STM8S.h.

6.18.2.137 PSCRH

```
struct { ...  } PSCRH
```

TIM1 16-bit prescaler high byte (TIM1_PSCRH)

6.18.2.138 PSCRL

```
struct { ...  } PSCRL
```

TIM1 16-bit prescaler low byte (TIM1_PSCRL)

6.18.2.139 RCR

```
struct { ...  } RCR
```

TIM1 Repetition counter (TIM1_RCR)

6.18.2.140 REP

_BITS REP

Repetition counter value.

Definition at line 2718 of file STM8AF_STM8S.h.

6.18.2.141 SMCR

struct { ... } SMCR

TIM1 Slave mode control register (TIM1_SMCR)

6.18.2.142 SMS

_BITS SMS

Clock/trigger/slave mode selection.

Definition at line 2501 of file STM8AF_STM8S.h.

6.18.2.143 SR1

struct { ... } SR1

TIM1 Status register 1 (TIM1_SR1)

6.18.2.144 SR2

struct { ... } SR2

TIM1 Status register 2 (TIM1_SR2)

6.18.2.145 TG

_BITS TG

Trigger generation.

Definition at line 2562 of file STM8AF_STM8S.h.

6.18.2.146 TIE

_BITS TIE

Trigger interrupt enable.

Definition at line 2525 of file STM8AF_STM8S.h.

6.18.2.147 TIF

_BITS TIF

Trigger interrupt flag.

Definition at line 2538 of file STM8AF_STM8S.h.

6.18.2.148 TS

_BITS TS

Trigger selection.

Definition at line 2503 of file STM8AF_STM8S.h.

6.18.2.149 UDIS

_BITS UDIS

Update disable.

Definition at line 2479 of file STM8AF_STM8S.h.

6.18.2.150 UG

_BITS UG

Update generation.

Definition at line 2556 of file STM8AF_STM8S.h.

6.18.2.151 UIE

_BITS UIE

Update interrupt enable.

Definition at line 2519 of file STM8AF_STM8S.h.

6.18.2.152 UIF

_BITS UIF

Update interrupt flag.

Definition at line 2532 of file STM8AF_STM8S.h.

6.18.2.153 URS

_BITS URS

Update request source.

Definition at line 2480 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

6.19 TIM2_3_t Struct Reference

struct for controlling 16-Bit Timer 2+3 (TIM2, TIM3)

```
#include <STM8L10x.h>
```

Data Fields

- struct {
  _BITS CEN: 1
  Counter enable.
  _BITS UDIS: 1
  Update disable.
  _BITS URS: 1
  Update request source.
  _BITS OPM: 1
  One-pulse mode.
  _BITS DIR: 1
  Direction.
  _BITS __pad0__: 2
  _BITS ARPE: 1
  Auto-reload preload enable.
  } CR1

  TIM2/3 Control register 1 (TIM2/3_CR1)
- struct {
  _BITS __pad0__: 4
  _BITS MMS: 3
  Master mode selection.
  _BITS __pad1__: 1
  } CR2

  TIM2/3 Control register 2 (TIM2/3_CR2)
- struct {
  _BITS SMS: 3
  Clock/trigger/slave mode selection.
  _BITS __pad0__: 1
  _BITS TS: 3
  Trigger selection.
  _BITS MSM: 1
  Master/slave mode.
  } SMCR

  TIM2/3 Slave mode control register (TIM2/3_SMCR)
- struct {
  _BITS ETF: 4
  External trigger filter.
  _BITS ETPS: 2
  External trigger prescaler.
  _BITS ECE: 1
  External clock enable.
  _BITS ETP: 1
  External trigger polarity.
  } ETR

  TIM2/3 External trigger register (TIM2/3_ETR)
- struct {
  _BITS UIE: 1
  Update interrupt enable.
  _BITS CC1IE: 1
  Capture/compare 1 interrupt enable.
  _BITS CC2IE: 1
  Capture/compare 2 interrupt enable.

_BITS __pad0__: 3
_BITS TIE: 1
Trigger interrupt enable.
_BITS BIE: 1
Break interrupt enable.
} IER

TIM2/3 Interrupt enable register (TIM2/3_IER)

- struct {
_BITS UIF: 1
Update interrupt flag.
_BITS CC1IF: 1
Capture/compare 1 interrupt flag.
_BITS CC2IF: 1
Capture/compare 2 interrupt flag.
_BITS __pad0__: 3
_BITS TIF: 1
Trigger interrupt flag.
_BITS BIF: 1
Break interrupt flag.
} SR1

TIM2/3 Status register 1 (TIM2/3_SR1)

- struct {
_BITS __pad0__: 1
_BITS CC1OF: 1
Capture/compare 1 overcapture flag.
_BITS CC2OF: 1
Capture/compare 2 overcapture flag.
_BITS __pad1__: 5
} SR2

TIM2/3 Status register 2 (TIM2/3_SR2)

- struct {
_BITS UG: 1
Update generation.
_BITS CC1G: 1
Capture/compare 1 generation.
_BITS CC2G: 1
Capture/compare 2 generation.
_BITS __pad0__: 3
_BITS TG: 1
Trigger generation.
_BITS BG: 1
Break generation.
} EGR

TIM2/3 Event generation register (TIM2/3_EGR)

- union {
struct {
_BITS CC1S: 2
Compare 1 selection.
_BITS OC1FE: 1
Output compare 1 fast enable.
_BITS OC1PE: 1
Output compare 1 preload enable.
_BITS OC1M: 3

Output compare 1 mode.
_BITS __pad0__: 1
} OUT
bitwise access to register (output mode)
struct {
_BITS CC1S: 2
Capture 1 selection.
_BITS IC1PSC: 2
Input capture 1 prescaler.
_BITS IC1F: 4
Input capture 1 filter.
} IN
bitwise access to register (input mode)
} CCMR1

TIM2/3 Capture/compare mode register 1 (TIM2/3_CCMR1)

- union {
struct {
_BITS CC2S: 2
Compare 2 selection.
_BITS OC2FE: 1
Output compare 2 fast enable.
_BITS OC2PE: 1
Output compare 2 preload enable.
_BITS OC2M: 3
Output compare 2 mode.
_BITS __pad0__: 1
} OUT
bitwise access to register (output mode)
struct {
_BITS CC2S: 2
Capture/compare 2 selection.
_BITS IC2PSC: 2
Input capture 2 prescaler.
_BITS IC2F: 4
Input capture 2 filter.
} IN
bitwise access to register (input mode)
} CCMR2

TIM2/3 Capture/compare mode register 2 (TIM2/3_CCMR2)

- struct {
_BITS CC1E: 1
Capture/compare 1 output enable.
_BITS CC1P: 1
Capture/compare 1 output polarity.
_BITS __pad0__: 2
_BITS CC2E: 1
Capture/compare 2 output enable.
_BITS CC2P: 1
Capture/compare 2 output polarity.
_BITS __pad1__: 2
} CCER1

TIM2/3 Capture/compare enable register 1 (TIM2/3_CCER1)

- struct {
_BITS CNT: 8
16-bit counter [15:8]

} CNTRH

TIM2/3 16-bit counter high byte (TIM2/3_CNTRH)

- struct {
  _BITS CNT: 8
  16-bit counter [7:0]
  } CNTRL

TIM2/3 16-bit counter low byte (TIM2/3_CNTRL)

- struct {
  _BITS PSC: 3
  prescaler [2:0]
  _BITS __pad0__: 5
  } PSCR

TIM2/3 16-bit prescaler (TIM2/3_PSCR)

- struct {
  _BITS ARR: 8
  16-bit auto-reload value [15:8]
  } ARRH

TIM2/3 16-bit auto-reload value high byte (TIM2/3_ARRH)

- struct {
  _BITS ARR: 8
  16-bit auto-reload value [7:0]
  } ARRL

TIM2/3 16-bit auto-reload value low byte (TIM2/3_ARRL)

- struct {
  _BITS CCR1: 8
  16-bit capture/compare value 1 [15:8]
  } CCR1H

TIM2/3 16-bit capture/compare value 1 high byte (TIM2/3_CCR1H)

- struct {
  _BITS CCR1: 8
  16-bit capture/compare value 1 [7:0]
  } CCR1L

TIM2/3 16-bit capture/compare value 1 low byte (TIM2/3_CCR1L)

- struct {
  _BITS CCR2: 8
  16-bit capture/compare value 2 [15:8]
  } CCR2H

TIM2/3 16-bit capture/compare value 2 high byte (TIM2/3_CCR2H)

- struct {
  _BITS CCR2: 8
  16-bit capture/compare value 2 [7:0]
  } CCR2L

TIM2/3 16-bit capture/compare value 2 low byte (TIM2/3_CCR2L)

- struct {
  _BITS LOCK: 2
  Lock configuration.
  _BITS OSSI: 1

Off state selection for idle mode.
_BITS __pad0__: 1
Reserved.
_BITS BKE: 1
Break enable.
_BITS BKP: 1
Break polarity.
_BITS AOE: 1
Automatic output enable.
_BITS MOE: 1
Main output enable.
} BKR

TIM2/3 Break register (TIM2/3_BKR)

- struct {
_BITS OIS1: 1
Output idle state 1 (OC1 output)
_BITS __pad0__: 1
Reserved.
_BITS OIS2: 1
Output idle state 2 (OC2 output)
_BITS __pad1__: 5
} OISR

TIM2/3 Output idle state register (TIM2/3_OISR)

6.19.1 Detailed Description

struct for controlling 16-Bit Timer 2+3 (TIM2, TIM3)

Definition at line 1553 of file STM8L10x.h.

6.19.2 Field Documentation

6.19.2.1 __pad0__

_BITS __pad0__

Reserved.

Definition at line 1562 of file STM8L10x.h.

6.19.2.2 __pad1__

_BITS __pad1__

Definition at line 1571 of file STM8L10x.h.

6.19.2.3 AOE

_BITS AOE

Automatic output enable.

Definition at line 1752 of file STM8L10x.h.

6.19.2.4 ARPE

_BITS ARPE

Auto-reload preload enable.

Definition at line 1563 of file STM8L10x.h.

6.19.2.5 ARR

_BITS ARR

16-bit auto-reload value [15:8]

16-bit auto-reload value [7:0]

Definition at line 1711 of file STM8L10x.h.

6.19.2.6 ARRH

struct { ... } ARRH

TIM2/3 16-bit auto-reload value high byte (TIM2/3_ARRH)

6.19.2.7 ARRL

struct { ... } ARRL

TIM2/3 16-bit auto-reload value low byte (TIM2/3_ARRL)

6.19.2.8 BG

_BITS BG

Break generation.

Definition at line 1631 of file STM8L10x.h.

6.19.2.9 BIE

_BITS BIE

Break interrupt enable.

Definition at line 1600 of file STM8L10x.h.

6.19.2.10 BIF

_BITS BIF

Break interrupt flag.

Definition at line 1611 of file STM8L10x.h.

6.19.2.11 BKE

_BITS BKE

Break enable.

Definition at line 1750 of file STM8L10x.h.

6.19.2.12 BKP

_BITS BKP

Break polarity.

Definition at line 1751 of file STM8L10x.h.

6.19.2.13 BKR

```
struct { ...  } BKR
```

TIM2/3 Break register (TIM2/3_BKR)

6.19.2.14 CC1E

```
_BITS CC1E
```

Capture/compare 1 output enable.

Definition at line 1681 of file STM8L10x.h.

6.19.2.15 CC1G

```
_BITS CC1G
```

Capture/compare 1 generation.

Definition at line 1627 of file STM8L10x.h.

6.19.2.16 CC1IE

```
_BITS CC1IE
```

Capture/compare 1 interrupt enable.

Definition at line 1596 of file STM8L10x.h.

6.19.2.17 CC1IF

```
_BITS CC1IF
```

Capture/compare 1 interrupt flag.

Definition at line 1607 of file STM8L10x.h.

6.19.2.18 CC1OF

_BITS CC1OF

Capture/compare 1 overcapture flag.

Definition at line 1618 of file STM8L10x.h.

6.19.2.19 CC1P

_BITS CC1P

Capture/compare 1 output polarity.

Definition at line 1682 of file STM8L10x.h.

6.19.2.20 CC1S

_BITS CC1S

Compare 1 selection.

Capture 1 selection.

Definition at line 1640 of file STM8L10x.h.

6.19.2.21 CC2E

_BITS CC2E

Capture/compare 2 output enable.

Definition at line 1684 of file STM8L10x.h.

6.19.2.22 CC2G

_BITS CC2G

Capture/compare 2 generation.

Definition at line 1628 of file STM8L10x.h.

6.19.2.23 CC2IE

_BITS CC2IE

Capture/compare 2 interrupt enable.

Definition at line 1597 of file STM8L10x.h.

6.19.2.24 CC2IF

_BITS CC2IF

Capture/compare 2 interrupt flag.

Definition at line 1608 of file STM8L10x.h.

6.19.2.25 CC2OF

_BITS CC2OF

Capture/compare 2 overcapture flag.

Definition at line 1619 of file STM8L10x.h.

6.19.2.26 CC2P

_BITS CC2P

Capture/compare 2 output polarity.

Definition at line 1685 of file STM8L10x.h.

6.19.2.27 CC2S

_BITS CC2S

Compare 2 selection.

Capture/compare 2 selection.

Definition at line 1662 of file STM8L10x.h.

6.19.2.28 CCER1

```
struct { ...  } CCER1
```

TIM2/3 Capture/compare enable register 1 (TIM2/3_CCER1)

6.19.2.29 CCMR1

```
union { ...  } CCMR1
```

TIM2/3 Capture/compare mode register 1 (TIM2/3_CCMR1)

6.19.2.30 CCMR2

```
union { ...  } CCMR2
```

TIM2/3 Capture/compare mode register 2 (TIM2/3_CCMR2)

6.19.2.31 CCR1

```
_BITS CCR1
```

16-bit capture/compare value 1 [15:8]

16-bit capture/compare value 1 [7:0]

Definition at line 1723 of file STM8L10x.h.

6.19.2.32 CCR1H

```
struct { ...  } CCR1H
```

TIM2/3 16-bit capture/compare value 1 high byte (TIM2/3_CCR1H)

6.19.2.33 CCR1L

```
struct { ...  } CCR1L
```

TIM2/3 16-bit capture/compare value 1 low byte (TIM2/3_CCR1L)

6.19.2.34 CCR2

_BITS CCR2

16-bit capture/compare value 2 [15:8]

16-bit capture/compare value 2 [7:0]

Definition at line 1735 of file STM8L10x.h.

6.19.2.35 CCR2H

struct { ... } CCR2H

TIM2/3 16-bit capture/compare value 2 high byte (TIM2/3_CCR2H)

6.19.2.36 CCR2L

struct { ... } CCR2L

TIM2/3 16-bit capture/compare value 2 low byte (TIM2/3_CCR2L)

6.19.2.37 CEN

_BITS CEN

Counter enable.

Definition at line 1557 of file STM8L10x.h.

6.19.2.38 CNT

_BITS CNT

16-bit counter [15:8]

16-bit counter [7:0]

Definition at line 1692 of file STM8L10x.h.

6.19.2.39 CNTRH

```
struct { ...  } CNTRH
```

TIM2/3 16-bit counter high byte (TIM2/3_CNTRH)

6.19.2.40 CNTRL

```
struct { ...  } CNTRL
```

TIM2/3 16-bit counter low byte (TIM2/3_CNTRL)

6.19.2.41 CR1

```
struct { ...  } CR1
```

TIM2/3 Control register 1 (TIM2/3_CR1)

6.19.2.42 CR2

```
struct { ...  } CR2
```

TIM2/3 Control register 2 (TIM2/3_CR2)

6.19.2.43 DIR

```
_BITS DIR
```

Direction.

Definition at line 1561 of file STM8L10x.h.

6.19.2.44 ECE

```
_BITS ECE
```

External clock enable.

Definition at line 1588 of file STM8L10x.h.

6.19.2.45 EGR

```
struct { ...  } EGR
```

TIM2/3 Event generation register (TIM2/3_EGR)

6.19.2.46 ETF

```
_BITS ETF
```

External trigger filter.

Definition at line 1586 of file STM8L10x.h.

6.19.2.47 ETP

```
_BITS ETP
```

External trigger polarity.

Definition at line 1589 of file STM8L10x.h.

6.19.2.48 ETPS

```
_BITS ETPS
```

External trigger prescaler.

Definition at line 1587 of file STM8L10x.h.

6.19.2.49 ETR

```
struct { ...  } ETR
```

TIM2/3 External trigger register (TIM2/3_ETR)

6.19.2.50 IC1F

_BITS IC1F

Input capture 1 filter.

Definition at line 1651 of file STM8L10x.h.

6.19.2.51 IC1PSC

_BITS IC1PSC

Input capture 1 prescaler.

Definition at line 1650 of file STM8L10x.h.

6.19.2.52 IC2F

_BITS IC2F

Input capture 2 filter.

Definition at line 1673 of file STM8L10x.h.

6.19.2.53 IC2PSC

_BITS IC2PSC

Input capture 2 prescaler.

Definition at line 1672 of file STM8L10x.h.

6.19.2.54 IER

struct { ... } IER

TIM2/3 Interrupt enable register (TIM2/3_IER)

6.19.2.55 IN [1/2]

```
struct { ...  } IN
```

bitwise access to register (input mode)

6.19.2.56 IN [2/2]

```
struct { ...  } IN
```

bitwise access to register (input mode)

6.19.2.57 LOCK

```
_BITS LOCK
```

Lock configuration.

Definition at line 1747 of file STM8L10x.h.

6.19.2.58 MMS

```
_BITS MMS
```

Master mode selection.

Definition at line 1570 of file STM8L10x.h.

6.19.2.59 MOE

```
_BITS MOE
```

Main output enable.

Definition at line 1753 of file STM8L10x.h.

6.19.2.60 MSM

_BITS MSM

Master/slave mode.

Definition at line 1580 of file STM8L10x.h.

6.19.2.61 OC1FE

_BITS OC1FE

Output compare 1 fast enable.

Definition at line 1641 of file STM8L10x.h.

6.19.2.62 OC1M

_BITS OC1M

Output compare 1 mode.

Definition at line 1643 of file STM8L10x.h.

6.19.2.63 OC1PE

_BITS OC1PE

Output compare 1 preload enable.

Definition at line 1642 of file STM8L10x.h.

6.19.2.64 OC2FE

_BITS OC2FE

Output compare 2 fast enable.

Definition at line 1663 of file STM8L10x.h.

6.19.2.65 OC2M

_BITS OC2M

Output compare 2 mode.

Definition at line 1665 of file STM8L10x.h.

6.19.2.66 OC2PE

_BITS OC2PE

Output compare 2 preload enable.

Definition at line 1664 of file STM8L10x.h.

6.19.2.67 OIS1

_BITS OIS1

Output idle state 1 (OC1 output)

Definition at line 1759 of file STM8L10x.h.

6.19.2.68 OIS2

_BITS OIS2

Output idle state 2 (OC2 output)

Definition at line 1761 of file STM8L10x.h.

6.19.2.69 OISR

struct { ... } OISR

TIM2/3 Output idle state register (TIM2/3_OISR)

6.19.2.70 OPM

```
_BITS OPM
```

One-pulse mode.

Definition at line 1560 of file STM8L10x.h.

6.19.2.71 OSSI

```
_BITS OSSI
```

Off state selection for idle mode.

Definition at line 1748 of file STM8L10x.h.

6.19.2.72 OUT [1/2]

```
struct { ...  } OUT
```

bitwise access to register (output mode)

6.19.2.73 OUT [2/2]

```
struct { ...  } OUT
```

bitwise access to register (output mode)

6.19.2.74 PSC

```
_BITS PSC
```

prescaler [2:0]

Definition at line 1704 of file STM8L10x.h.

6.19.2.75 PSCR

```
struct { ...  } PSCR
```

TIM2/3 16-bit prescaler (TIM2/3_PSCR)

6.19.2.76 SMCR

```
struct { ...  } SMCR
```

TIM2/3 Slave mode control register (TIM2/3_SMCR)

6.19.2.77 SMS

```
_BITS SMS
```

Clock/trigger/slave mode selection.

Definition at line 1577 of file STM8L10x.h.

6.19.2.78 SR1

```
struct { ...  } SR1
```

TIM2/3 Status register 1 (TIM2/3_SR1)

6.19.2.79 SR2

```
struct { ...  } SR2
```

TIM2/3 Status register 2 (TIM2/3_SR2)

6.19.2.80 TG

```
_BITS TG
```

Trigger generation.

Definition at line 1630 of file STM8L10x.h.

6.19.2.81 TIE

_BITS TIE

Trigger interrupt enable.

Definition at line 1599 of file STM8L10x.h.

6.19.2.82 TIF

_BITS TIF

Trigger interrupt flag.

Definition at line 1610 of file STM8L10x.h.

6.19.2.83 TS

_BITS TS

Trigger selection.

Definition at line 1579 of file STM8L10x.h.

6.19.2.84 UDIS

_BITS UDIS

Update disable.

Definition at line 1558 of file STM8L10x.h.

6.19.2.85 UG

_BITS UG

Update generation.

Definition at line 1626 of file STM8L10x.h.

6.19.2.86 UIE

_BITS UIE

Update interrupt enable.

Definition at line 1595 of file STM8L10x.h.

6.19.2.87 UIF

_BITS UIF

Update interrupt flag.

Definition at line 1606 of file STM8L10x.h.

6.19.2.88 URS

_BITS URS

Update request source.

Definition at line 1559 of file STM8L10x.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.20 TIM2_t Struct Reference

struct for controlling 16-Bit Timer 2 (TIM2)

#include <STM8AF_STM8S.h>

Data Fields

- struct {
  _BITS CEN: 1
  Counter enable.
  _BITS UDIS: 1
  Update disable.
  _BITS URS: 1
  Update request source.
  _BITS OPM: 1
  One-pulse mode.
  _BITS __pad0__: 3
  _BITS ARPE: 1
  Auto-reload preload enable.
  } CR1

  TIM2 Control register 1 (TIM2_CR1)
- struct {
  _BITS UIE: 1
  Update interrupt enable.
  _BITS CC1IE: 1
  Capture/compare 1 interrupt enable.
  _BITS CC2IE: 1
  Capture/compare 2 interrupt enable.
  _BITS CC3IE: 1
  Capture/compare 3 interrupt enable.
  _BITS __pad0__: 4
  } IER

  Reserved registers on selected devices (2B)
- struct {
  _BITS UIF: 1
  Update interrupt flag.
  _BITS CC1IF: 1
  Capture/compare 1 interrupt flag.
  _BITS CC2IF: 1
  Capture/compare 2 interrupt flag.
  _BITS CC3IF: 1
  Capture/compare 3 interrupt flag.
  _BITS __pad0__: 4
  } SR1

  TIM2 Status register 1 (TIM2_SR1)
- struct {
  _BITS __pad0__: 1
  _BITS CC1OF: 1
  Capture/compare 1 overcapture flag.
  _BITS CC2OF: 1
  Capture/compare 2 overcapture flag.
  _BITS CC3OF: 1
  Capture/compare 3 overcapture flag.
  _BITS __pad1__: 4
  } SR2

  TIM2 Status register 2 (TIM2_SR2)
- struct {
  _BITS UG: 1

Update generation.
_BITS CC1G: 1
Capture/compare 1 generation.
_BITS CC2G: 1
Capture/compare 2 generation.
_BITS CC3G: 1
Capture/compare 3 generation.
_BITS __pad0__: 4
} EGR

TIM2 Event generation register (TIM2_EGR)

- union {
  struct {
  _BITS CC1S: 2
  Compare 1 selection.
  _BITS __pad0__: 1
  _BITS OC1PE: 1
  Output compare 1 preload enable.
  _BITS OC1M: 3
  Output compare 1 mode.
  _BITS __pad1__: 1
  } OUT
  bitwise access to register (output mode)
  struct {
  _BITS CC1S: 2
  Capture 1 selection.
  _BITS IC1PSC: 2
  Input capture 1 prescaler.
  _BITS IC1F: 4
  Input capture 1 filter.
  } IN
  bitwise access to register (input mode)
  } CCMR1

TIM2 Capture/compare mode register 1 (TIM2_CCMR1)

- union {
  struct {
  _BITS CC2S: 2
  Capture/compare 2 selection.
  _BITS __pad0__: 1
  _BITS OC2PE: 1
  Output compare 2 preload enable.
  _BITS OC2M: 3
  Output compare 2 mode.
  _BITS __pad1__: 1
  } OUT
  bitwise access to register (output mode)
  struct {
  _BITS CC2S: 2
  Capture/compare 2 selection.
  _BITS IC2PSC: 2
  Input capture 2 prescaler.
  _BITS IC2F: 4
  Input capture 2 filter.
  } IN
  bitwise access to register (input mode)
  } CCMR2

TIM2 Capture/compare mode register 2 (TIM2_CCMR2)

- union {
  struct {
  _BITS CC3S: 2
  Capture/compare 3 selection.
  _BITS __pad0__: 1
  _BITS OC3PE: 1
  Output compare 3 preload enable.
  _BITS OC3M: 3
  Output compare 3 mode.
  _BITS OC3CE: 1
  Output compare 3 clear enable.
  } OUT
  bitwise access to register (output mode)
  struct {
  _BITS CC3S: 2
  Capture/compare 3 selection.
  _BITS IC3PSC: 2
  Input capture 3 prescaler.
  _BITS IC3F: 4
  Input capture 3 filter.
  } IN
  bitwise access to register (input mode)
  } CCMR3

TIM2 Capture/compare mode register 3 (TIM2_CCMR3)

- struct {
  _BITS CC1E: 1
  Capture/compare 1 output enable.
  _BITS CC1P: 1
  Capture/compare 1 output polarity.
  _BITS __pad0__: 2
  _BITS CC2E: 1
  Capture/compare 2 output enable.
  _BITS CC2P: 1
  Capture/compare 2 output polarity.
  _BITS __pad1__: 2
  } CCER1

TIM2 Capture/compare enable register 1 (TIM2_CCER1)

- struct {
  _BITS CC3E: 1
  Capture/compare 3 output enable.
  _BITS CC3P: 1
  Capture/compare 3 output polarity.
  _BITS __pad0__: 6
  } CCER2

TIM2 Capture/compare enable register 2 (TIM2_CCER2)

- struct {
  _BITS CNT: 8
  16-bit counter [15:8]
  } CNTRH

TIM2 16-bit counter high byte (TIM2_CNTRH)

- struct {
  _BITS CNT: 8

16-bit counter [7:0]
} CNTRL

TIM2 16-bit counter low byte (TIM2_CNTRL)

- struct {
  _BITS PSC: 4
  prescaler [3:0]
  _BITS __pad0__: 4
  } PSCR

TIM2 16-bit prescaler high byte (TIM2_PSCR)

- struct {
  _BITS ARR: 8
  16-bit auto-reload value [15:8]
  } ARRH

TIM2 16-bit auto-reload value high byte (TIM2_ARRH)

- struct {
  _BITS ARR: 8
  16-bit auto-reload value [7:0]
  } ARRL

TIM2 16-bit auto-reload value low byte (TIM2_ARRL)

- struct {
  _BITS CCR1: 8
  16-bit capture/compare value 1 [15:8]
  } CCR1H

TIM2 16-bit capture/compare value 1 high byte (TIM2_CCR1H)

- struct {
  _BITS CCR1: 8
  16-bit capture/compare value 1 [7:0]
  } CCR1L

TIM2 16-bit capture/compare value 1 low byte (TIM2_CCR1L)

- struct {
  _BITS CCR2: 8
  16-bit capture/compare value 2 [15:8]
  } CCR2H

TIM2 16-bit capture/compare value 2 high byte (TIM2_CCR2H)

- struct {
  _BITS CCR2: 8
  16-bit capture/compare value 2 [7:0]
  } CCR2L

TIM2 16-bit capture/compare value 2 low byte (TIM2_CCR2L)

- struct {
  _BITS CCR3: 8
  16-bit capture/compare value 3 [15:8]
  } CCR3H

TIM2 16-bit capture/compare value 3 high byte (TIM2_CCR3H)

- struct {
  _BITS CCR3: 8
  16-bit capture/compare value 3 [7:0]

} CCR3L

TIM2 16-bit capture/compare value 3 low byte (TIM2_CCR3L)

6.20.1 Detailed Description

struct for controlling 16-Bit Timer 2 (TIM2)

Definition at line 3097 of file STM8AF_STM8S.h.

6.20.2 Field Documentation

6.20.2.1 __pad0__

_BITS __pad0__

Definition at line 3105 of file STM8AF_STM8S.h.

6.20.2.2 __pad1__

_BITS __pad1__

Definition at line 3142 of file STM8AF_STM8S.h.

6.20.2.3 ARPE

_BITS ARPE

Auto-reload preload enable.

Definition at line 3106 of file STM8AF_STM8S.h.

6.20.2.4 ARR

_BITS ARR

16-bit auto-reload value [15:8]

16-bit auto-reload value [7:0]

Definition at line 3262 of file STM8AF_STM8S.h.

6.20.2.5 ARRH

```
struct { ...  } ARRH
```

TIM2 16-bit auto-reload value high byte (TIM2_ARRH)

6.20.2.6 ARRL

```
struct { ...  } ARRL
```

TIM2 16-bit auto-reload value low byte (TIM2_ARRL)

6.20.2.7 CC1E

```
_BITS CC1E
```

Capture/compare 1 output enable.

Definition at line 3224 of file STM8AF_STM8S.h.

6.20.2.8 CC1G

```
_BITS CC1G
```

Capture/compare 1 generation.

Definition at line 3149 of file STM8AF_STM8S.h.

6.20.2.9 CC1IE

```
_BITS CC1IE
```

Capture/compare 1 interrupt enable.

Definition at line 3119 of file STM8AF_STM8S.h.

6.20.2.10 CC1IF

_BITS CC1IF

Capture/compare 1 interrupt flag.

Definition at line 3129 of file STM8AF_STM8S.h.

6.20.2.11 CC1OF

_BITS CC1OF

Capture/compare 1 overcapture flag.

Definition at line 3139 of file STM8AF_STM8S.h.

6.20.2.12 CC1P

_BITS CC1P

Capture/compare 1 output polarity.

Definition at line 3225 of file STM8AF_STM8S.h.

6.20.2.13 CC1S

_BITS CC1S

Compare 1 selection.

Capture 1 selection.

Definition at line 3161 of file STM8AF_STM8S.h.

6.20.2.14 CC2E

_BITS CC2E

Capture/compare 2 output enable.

Definition at line 3227 of file STM8AF_STM8S.h.

6.20.2.15 CC2G

_BITS CC2G

Capture/compare 2 generation.

Definition at line 3150 of file STM8AF_STM8S.h.

6.20.2.16 CC2IE

_BITS CC2IE

Capture/compare 2 interrupt enable.

Definition at line 3120 of file STM8AF_STM8S.h.

6.20.2.17 CC2IF

_BITS CC2IF

Capture/compare 2 interrupt flag.

Definition at line 3130 of file STM8AF_STM8S.h.

6.20.2.18 CC2OF

_BITS CC2OF

Capture/compare 2 overcapture flag.

Definition at line 3140 of file STM8AF_STM8S.h.

6.20.2.19 CC2P

_BITS CC2P

Capture/compare 2 output polarity.

Definition at line 3228 of file STM8AF_STM8S.h.

6.20.2.20 CC2S

_BITS CC2S

Capture/compare 2 selection.

Definition at line 3183 of file STM8AF_STM8S.h.

6.20.2.21 CC3E

_BITS CC3E

Capture/compare 3 output enable.

Definition at line 3235 of file STM8AF_STM8S.h.

6.20.2.22 CC3G

_BITS CC3G

Capture/compare 3 generation.

Definition at line 3151 of file STM8AF_STM8S.h.

6.20.2.23 CC3IE

_BITS CC3IE

Capture/compare 3 interrupt enable.

Definition at line 3121 of file STM8AF_STM8S.h.

6.20.2.24 CC3IF

_BITS CC3IF

Capture/compare 3 interrupt flag.

Definition at line 3131 of file STM8AF_STM8S.h.

6.20.2.25 CC3OF

_BITS CC3OF

Capture/compare 3 overcapture flag.

Definition at line 3141 of file STM8AF_STM8S.h.

6.20.2.26 CC3P

_BITS CC3P

Capture/compare 3 output polarity.

Definition at line 3236 of file STM8AF_STM8S.h.

6.20.2.27 CC3S

_BITS CC3S

Capture/compare 3 selection.

Definition at line 3205 of file STM8AF_STM8S.h.

6.20.2.28 CCER1

struct { ... } CCER1

TIM2 Capture/compare enable register 1 (TIM2_CCER1)

6.20.2.29 CCER2

struct { ... } CCER2

TIM2 Capture/compare enable register 2 (TIM2_CCER2)

6.20.2.30 CCMR1

```
union { ... } CCMR1
```

TIM2 Capture/compare mode register 1 (TIM2_CCMR1)

6.20.2.31 CCMR2

```
union { ... } CCMR2
```

TIM2 Capture/compare mode register 2 (TIM2_CCMR2)

6.20.2.32 CCMR3

```
union { ... } CCMR3
```

TIM2 Capture/compare mode register 3 (TIM2_CCMR3)

6.20.2.33 CCR1

```
_BITS CCR1
```

16-bit capture/compare value 1 [15:8]

16-bit capture/compare value 1 [7:0]

Definition at line 3274 of file STM8AF_STM8S.h.

6.20.2.34 CCR1H

```
struct { ... } CCR1H
```

TIM2 16-bit capture/compare value 1 high byte (TIM2_CCR1H)

6.20.2.35 CCR1L

```
struct { ... } CCR1L
```

TIM2 16-bit capture/compare value 1 low byte (TIM2_CCR1L)

6.20.2.36 CCR2

_BITS CCR2

16-bit capture/compare value 2 [15:8]

16-bit capture/compare value 2 [7:0]

Definition at line 3286 of file STM8AF_STM8S.h.

6.20.2.37 CCR2H

struct { ... } CCR2H

TIM2 16-bit capture/compare value 2 high byte (TIM2_CCR2H)

6.20.2.38 CCR2L

struct { ... } CCR2L

TIM2 16-bit capture/compare value 2 low byte (TIM2_CCR2L)

6.20.2.39 CCR3

_BITS CCR3

16-bit capture/compare value 3 [15:8]

16-bit capture/compare value 3 [7:0]

Definition at line 3298 of file STM8AF_STM8S.h.

6.20.2.40 CCR3H

struct { ... } CCR3H

TIM2 16-bit capture/compare value 3 high byte (TIM2_CCR3H)

6.20.2.41 CCR3L

```
struct { ...  } CCR3L
```

TIM2 16-bit capture/compare value 3 low byte (TIM2_CCR3L)

6.20.2.42 CEN

```
_BITS CEN
```

Counter enable.

Definition at line 3101 of file STM8AF_STM8S.h.

6.20.2.43 CNT

```
_BITS CNT
```

16-bit counter [15:8]

16-bit counter [7:0]

Definition at line 3243 of file STM8AF_STM8S.h.

6.20.2.44 CNTRH

```
struct { ...  } CNTRH
```

TIM2 16-bit counter high byte (TIM2_CNTRH)

6.20.2.45 CNTRL

```
struct { ...  } CNTRL
```

TIM2 16-bit counter low byte (TIM2_CNTRL)

6.20.2.46 CR1

```
struct { ...  } CR1
```

TIM2 Control register 1 (TIM2_CR1)

6.20.2.47 EGR

```
struct { ...  } EGR
```

TIM2 Event generation register (TIM2_EGR)

6.20.2.48 IC1F

```
_BITS IC1F
```

Input capture 1 filter.

Definition at line 3172 of file STM8AF_STM8S.h.

6.20.2.49 IC1PSC

```
_BITS IC1PSC
```

Input capture 1 prescaler.

Definition at line 3171 of file STM8AF_STM8S.h.

6.20.2.50 IC2F

```
_BITS IC2F
```

Input capture 2 filter.

Definition at line 3194 of file STM8AF_STM8S.h.

6.20.2.51 IC2PSC

```
_BITS IC2PSC
```

Input capture 2 prescaler.

Definition at line 3193 of file STM8AF_STM8S.h.

6.20.2.52 IC3F

```
_BITS IC3F
```

Input capture 3 filter.

Definition at line 3216 of file STM8AF_STM8S.h.

6.20.2.53 IC3PSC

```
_BITS IC3PSC
```

Input capture 3 prescaler.

Definition at line 3215 of file STM8AF_STM8S.h.

6.20.2.54 IER

```
struct { ...  } IER
```

Reserved registers on selected devices (2B)

TIM2 Interrupt enable register (TIM2_IER)

6.20.2.55 IN [1/3]

```
struct { ...  } IN
```

bitwise access to register (input mode)

6.20.2.56 IN [2/3]

```
struct { ...  } IN
```

bitwise access to register (input mode)

6.20.2.57 IN [3/3]

```
struct { ...  } IN
```

bitwise access to register (input mode)

6.20.2.58 OC1M

_BITS OC1M

Output compare 1 mode.

Definition at line 3164 of file STM8AF_STM8S.h.

6.20.2.59 OC1PE

_BITS OC1PE

Output compare 1 preload enable.

Definition at line 3163 of file STM8AF_STM8S.h.

6.20.2.60 OC2M

_BITS OC2M

Output compare 2 mode.

Definition at line 3186 of file STM8AF_STM8S.h.

6.20.2.61 OC2PE

_BITS OC2PE

Output compare 2 preload enable.

Definition at line 3185 of file STM8AF_STM8S.h.

6.20.2.62 OC3CE

_BITS OC3CE

Output compare 3 clear enable.

Definition at line 3209 of file STM8AF_STM8S.h.

6.20.2.63 OC3M

_BITS OC3M

Output compare 3 mode.

Definition at line 3208 of file STM8AF_STM8S.h.

6.20.2.64 OC3PE

_BITS OC3PE

Output compare 3 preload enable.

Definition at line 3207 of file STM8AF_STM8S.h.

6.20.2.65 OPM

_BITS OPM

One-pulse mode.

Definition at line 3104 of file STM8AF_STM8S.h.

6.20.2.66 OUT [1/3]

```
struct { ...  } OUT
```

bitwise access to register (output mode)

6.20.2.67 OUT [2/3]

```
struct { ...  } OUT
```

bitwise access to register (output mode)

6.20.2.68 OUT [3/3]

```
struct { ...  } OUT
```

bitwise access to register (output mode)

6.20.2.69 PSC

```
_BITS PSC
```

prescaler [3:0]

Definition at line 3255 of file STM8AF_STM8S.h.

6.20.2.70 PSCR

```
struct { ...  } PSCR
```

TIM2 16-bit prescaler high byte (TIM2_PSCR)

6.20.2.71 SR1

```
struct { ...  } SR1
```

TIM2 Status register 1 (TIM2_SR1)

6.20.2.72 SR2

```
struct { ...  } SR2
```

TIM2 Status register 2 (TIM2_SR2)

6.20.2.73 UDIS

```
_BITS UDIS
```

Update disable.

Definition at line 3102 of file STM8AF_STM8S.h.

6.20.2.74 UG

```
_BITS UG
```

Update generation.

Definition at line 3148 of file STM8AF_STM8S.h.

6.20.2.75 UIE

```
_BITS UIE
```

Update interrupt enable.

Definition at line 3118 of file STM8AF_STM8S.h.

6.20.2.76 UIF

```
_BITS UIF
```

Update interrupt flag.

Definition at line 3128 of file STM8AF_STM8S.h.

6.20.2.77 URS

_BITS URS

Update request source.

Definition at line 3103 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

6.21 TIM3_t Struct Reference

struct for controlling 16-Bit Timer 3 (TIM3)

`#include <STM8AF_STM8S.h>`

Data Fields

- struct {
  _BITS CEN: 1
  Counter enable.
  _BITS UDIS: 1
  Update disable.
  _BITS URS: 1
  Update request source.
  _BITS OPM: 1
  One-pulse mode.
  _BITS __pad0__: 3
  _BITS ARPE: 1
  Auto-reload preload enable.
  } CR1

  TIM3 Control register 1 (TIM3_CR1)
- struct {
  _BITS UIE: 1
  Update interrupt enable.
  _BITS CC1IE: 1
  Capture/compare 1 interrupt enable.
  _BITS CC2IE: 1
  Capture/compare 2 interrupt enable.
  _BITS __pad0__: 5
  } IER

  TIM3 Interrupt enable register (TIM3_IER)
- struct {
  _BITS UIF: 1
  Update interrupt flag.
  _BITS CC1IF: 1
  Capture/compare 1 interrupt flag.
  _BITS CC2IF: 1
  Capture/compare 2 interrupt flag.
  _BITS __pad0__: 5
  } SR1

TIM3 Status register 1 (TIM3_SR1)

- struct {
  _BITS __pad0__: 1
  _BITS CC1OF: 1
  Capture/compare 1 overcapture flag.
  _BITS CC2OF: 1
  Capture/compare 2 overcapture flag.
  _BITS __pad1__: 5
  } SR2

TIM3 Status register 2 (TIM3_SR2)

- struct {
  _BITS UG: 1
  Update generation.
  _BITS CC1G: 1
  Capture/compare 1 generation.
  _BITS CC2G: 1
  Capture/compare 2 generation.
  _BITS __pad0__: 5
  } EGR

TIM3 Event generation register (TIM3_EGR)

- union {
  struct {
  _BITS CC1S: 2
  Compare 1 selection.
  _BITS __pad0__: 1
  _BITS OC1PE: 1
  Output compare 1 preload enable.
  _BITS OC1M: 3
  Output compare 1 mode.
  _BITS __pad1__: 1
  } OUT
  bitwise access to register (output mode)
  struct {
  _BITS CC1S: 2
  Capture 1 selection.
  _BITS IC1PSC: 2
  Input capture 1 prescaler.
  _BITS IC1F: 4
  Input capture 1 filter.
  } IN
  bitwise access to register (input mode)
  } CCMR1

TIM3 Capture/compare mode register 1 (TIM3_CCMR1)

- union {
  struct {
  _BITS CC2S: 2
  Capture/compare 2 selection.
  _BITS __pad0__: 1
  _BITS OC2PE: 1
  Output compare 2 preload enable.
  _BITS OC2M: 3
  Output compare 2 mode.
  _BITS __pad1__: 1
  } OUT

bitwise access to register (output mode)
struct {
_BITS CC2S: 2
Capture/compare 2 selection.
_BITS IC2PSC: 2
Input capture 2 prescaler.
_BITS IC2F: 4
Input capture 2 filter.
} IN
bitwise access to register (input mode)
} CCMR2

TIM3 Capture/compare mode register 2 (TIM3_CCMR2)

- struct {
_BITS CC1E: 1
Capture/compare 1 output enable.
_BITS CC1P: 1
Capture/compare 1 output polarity.
_BITS __pad0__: 2
_BITS CC2E: 1
Capture/compare 2 output enable.
_BITS CC2P: 1
Capture/compare 2 output polarity.
_BITS __pad1__: 2
} CCER1

TIM3 Capture/compare enable register 1 (TIM3_CCER1)

- struct {
_BITS CNT: 8
16-bit counter [15:8]
} CNTRH

TIM3 16-bit counter high byte (TIM3_CNTRH)

- struct {
_BITS CNT: 8
16-bit counter [7:0]
} CNTRL

TIM3 16-bit counter low byte (TIM3_CNTRL)

- struct {
_BITS PSC: 4
prescaler [3:0]
_BITS __pad0__: 4
} PSCR

TIM3 16-bit prescaler high byte (TIM3_PSCR)

- struct {
_BITS ARR: 8
16-bit auto-reload value [15:8]
} ARRH

TIM3 16-bit auto-reload value high byte (TIM3_ARRH)

- struct {
_BITS ARR: 8
16-bit auto-reload value [7:0]
} ARRL

TIM3 16-bit auto-reload value low byte (TIM3_ARRL)

- struct {
  _BITS CCR1: 8
  16-bit capture/compare value 1 [15:8]
  } CCR1H

TIM3 16-bit capture/compare value 1 high byte (TIM3_CCR1H)

- struct {
  _BITS CCR1: 8
  16-bit capture/compare value 1 [7:0]
  } CCR1L

TIM3 16-bit capture/compare value 1 low byte (TIM3_CCR1L)

- struct {
  _BITS CCR2: 8
  16-bit capture/compare value 2 [15:8]
  } CCR2H

TIM3 16-bit capture/compare value 2 high byte (TIM3_CCR2H)

- struct {
  _BITS CCR2: 8
  16-bit capture/compare value 2 [7:0]
  } CCR2L

TIM3 16-bit capture/compare value 2 low byte (TIM3_CCR2L)

6.21.1 Detailed Description

struct for controlling 16-Bit Timer 3 (TIM3)

Definition at line 3516 of file STM8AF_STM8S.h.

6.21.2 Field Documentation

6.21.2.1 __pad0__

```
_BITS __pad0__
```

Definition at line 3524 of file STM8AF_STM8S.h.

6.21.2.2 __pad1__

```
_BITS __pad1__
```

Definition at line 3552 of file STM8AF_STM8S.h.

6.21.2.3 ARPE

_BITS ARPE

Auto-reload preload enable.

Definition at line 3525 of file STM8AF_STM8S.h.

6.21.2.4 ARR

_BITS ARR

16-bit auto-reload value [15:8]

16-bit auto-reload value [7:0]

Definition at line 3641 of file STM8AF_STM8S.h.

6.21.2.5 ARRH

struct { ... } ARRH

TIM3 16-bit auto-reload value high byte (TIM3_ARRH)

6.21.2.6 ARRL

struct { ... } ARRL

TIM3 16-bit auto-reload value low byte (TIM3_ARRL)

6.21.2.7 CC1E

_BITS CC1E

Capture/compare 1 output enable.

Definition at line 3611 of file STM8AF_STM8S.h.

6.21.2.8 CC1G

_BITS CC1G

Capture/compare 1 generation.

Definition at line 3559 of file STM8AF_STM8S.h.

6.21.2.9 CC1IE

_BITS CC1IE

Capture/compare 1 interrupt enable.

Definition at line 3532 of file STM8AF_STM8S.h.

6.21.2.10 CC1IF

_BITS CC1IF

Capture/compare 1 interrupt flag.

Definition at line 3541 of file STM8AF_STM8S.h.

6.21.2.11 CC1OF

_BITS CC1OF

Capture/compare 1 overcapture flag.

Definition at line 3550 of file STM8AF_STM8S.h.

6.21.2.12 CC1P

_BITS CC1P

Capture/compare 1 output polarity.

Definition at line 3612 of file STM8AF_STM8S.h.

6.21.2.13 CC1S

_BITS CC1S

Compare 1 selection.

Capture 1 selection.

Definition at line 3570 of file STM8AF_STM8S.h.

6.21.2.14 CC2E

_BITS CC2E

Capture/compare 2 output enable.

Definition at line 3614 of file STM8AF_STM8S.h.

6.21.2.15 CC2G

_BITS CC2G

Capture/compare 2 generation.

Definition at line 3560 of file STM8AF_STM8S.h.

6.21.2.16 CC2IE

_BITS CC2IE

Capture/compare 2 interrupt enable.

Definition at line 3533 of file STM8AF_STM8S.h.

6.21.2.17 CC2IF

_BITS CC2IF

Capture/compare 2 interrupt flag.

Definition at line 3542 of file STM8AF_STM8S.h.

6.21.2.18 CC2OF

```
_BITS CC2OF
```

Capture/compare 2 overcapture flag.

Definition at line 3551 of file STM8AF_STM8S.h.

6.21.2.19 CC2P

```
_BITS CC2P
```

Capture/compare 2 output polarity.

Definition at line 3615 of file STM8AF_STM8S.h.

6.21.2.20 CC2S

```
_BITS CC2S
```

Capture/compare 2 selection.

Definition at line 3592 of file STM8AF_STM8S.h.

6.21.2.21 CCER1

```
struct { ...  } CCER1
```

TIM3 Capture/compare enable register 1 (TIM3_CCER1)

6.21.2.22 CCMR1

```
union { ...  } CCMR1
```

TIM3 Capture/compare mode register 1 (TIM3_CCMR1)

6.21.2.23 CCMR2

```
union { ... } CCMR2
```

TIM3 Capture/compare mode register 2 (TIM3_CCMR2)

6.21.2.24 CCR1

```
_BITS CCR1
```

16-bit capture/compare value 1 [15:8]

16-bit capture/compare value 1 [7:0]

Definition at line 3653 of file STM8AF_STM8S.h.

6.21.2.25 CCR1H

```
struct { ... } CCR1H
```

TIM3 16-bit capture/compare value 1 high byte (TIM3_CCR1H)

6.21.2.26 CCR1L

```
struct { ... } CCR1L
```

TIM3 16-bit capture/compare value 1 low byte (TIM3_CCR1L)

6.21.2.27 CCR2

```
_BITS CCR2
```

16-bit capture/compare value 2 [15:8]

16-bit capture/compare value 2 [7:0]

Definition at line 3665 of file STM8AF_STM8S.h.

6.21.2.28 CCR2H

```
struct { ...  } CCR2H
```

TIM3 16-bit capture/compare value 2 high byte (TIM3_CCR2H)

6.21.2.29 CCR2L

```
struct { ...  } CCR2L
```

TIM3 16-bit capture/compare value 2 low byte (TIM3_CCR2L)

6.21.2.30 CEN

```
_BITS CEN
```

Counter enable.

Definition at line 3520 of file STM8AF_STM8S.h.

6.21.2.31 CNT

```
_BITS CNT
```

16-bit counter [15:8]

16-bit counter [7:0]

Definition at line 3622 of file STM8AF_STM8S.h.

6.21.2.32 CNTRH

```
struct { ...  } CNTRH
```

TIM3 16-bit counter high byte (TIM3_CNTRH)

6.21.2.33 CNTRL

```
struct { ...  } CNTRL
```

TIM3 16-bit counter low byte (TIM3_CNTRL)

6.21.2.34 CR1

```
struct { ...  } CR1
```

TIM3 Control register 1 (TIM3_CR1)

6.21.2.35 EGR

```
struct { ...  } EGR
```

TIM3 Event generation register (TIM3_EGR)

6.21.2.36 IC1F

```
_BITS IC1F
```

Input capture 1 filter.

Definition at line 3581 of file STM8AF_STM8S.h.

6.21.2.37 IC1PSC

```
_BITS IC1PSC
```

Input capture 1 prescaler.

Definition at line 3580 of file STM8AF_STM8S.h.

6.21.2.38 IC2F

```
_BITS IC2F
```

Input capture 2 filter.

Definition at line 3603 of file STM8AF_STM8S.h.

6.21.2.39 IC2PSC

_BITS IC2PSC

Input capture 2 prescaler.

Definition at line 3602 of file STM8AF_STM8S.h.

6.21.2.40 IER

struct { ... } IER

TIM3 Interrupt enable register (TIM3_IER)

6.21.2.41 IN [1/2]

struct { ... } IN

bitwise access to register (input mode)

6.21.2.42 IN [2/2]

struct { ... } IN

bitwise access to register (input mode)

6.21.2.43 OC1M

_BITS OC1M

Output compare 1 mode.

Definition at line 3573 of file STM8AF_STM8S.h.

6.21.2.44 OC1PE

_BITS OC1PE

Output compare 1 preload enable.

Definition at line 3572 of file STM8AF_STM8S.h.

6.21.2.45 OC2M

_BITS OC2M

Output compare 2 mode.

Definition at line 3595 of file STM8AF_STM8S.h.

6.21.2.46 OC2PE

_BITS OC2PE

Output compare 2 preload enable.

Definition at line 3594 of file STM8AF_STM8S.h.

6.21.2.47 OPM

_BITS OPM

One-pulse mode.

Definition at line 3523 of file STM8AF_STM8S.h.

6.21.2.48 OUT [1/2]

struct { ... } OUT

bitwise access to register (output mode)

6.21.2.49 OUT [2/2]

struct { ... } OUT

bitwise access to register (output mode)

6.21.2.50 PSC

```
_BITS PSC
```

prescaler [3:0]

Definition at line 3634 of file STM8AF_STM8S.h.

6.21.2.51 PSCR

```
struct { ...  } PSCR
```

TIM3 16-bit prescaler high byte (TIM3_PSCR)

6.21.2.52 SR1

```
struct { ...  } SR1
```

TIM3 Status register 1 (TIM3_SR1)

6.21.2.53 SR2

```
struct { ...  } SR2
```

TIM3 Status register 2 (TIM3_SR2)

6.21.2.54 UDIS

```
_BITS UDIS
```

Update disable.

Definition at line 3521 of file STM8AF_STM8S.h.

6.21.2.55 UG

```
_BITS UG
```

Update generation.

Definition at line 3558 of file STM8AF_STM8S.h.

6.21.2.56 UIE

_BITS UIE

Update interrupt enable.

Definition at line 3531 of file STM8AF_STM8S.h.

6.21.2.57 UIF

_BITS UIF

Update interrupt flag.

Definition at line 3540 of file STM8AF_STM8S.h.

6.21.2.58 URS

_BITS URS

Update request source.

Definition at line 3522 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

6.22 TIM4_t Struct Reference

struct for controlling 8-Bit Timer 4 (TIM4)

#include <STM8AF_STM8S.h>

Data Fields

- struct {
  _BITS CEN: 1
  Counter enable.
  _BITS UDIS: 1
  Update disable.
  _BITS URS: 1
  Update request source.
  _BITS OPM: 1
  One-pulse mode.
  _BITS __pad0__: 3
  _BITS ARPE: 1
  Auto-reload preload enable.
  } CR

  TIM4 Control register (TIM4_CR)
- struct {
  _BITS UIE: 1
  Update interrupt enable.
  _BITS __pad0__: 7
  } IER

  Reserved registers on selected devices (2B)
- struct {
  _BITS UIF: 1
  Update interrupt flag.
  _BITS __pad0__: 7
  } SR

  TIM4 Status register (TIM4_SR)
- struct {
  _BITS UG: 1
  Update generation.
  _BITS __pad0__: 7
  } EGR

  TIM4 Event Generation (TIM4_EGR)
- struct {
  _BITS CNT: 8
  8-bit counter
  } CNTR

  TIM4 8-bit counter register (TIM4_CNTR)
- struct {
  _BITS PSC: 3
  clock prescaler
  _BITS __pad0__: 5
  } PSCR

  TIM4 clock prescaler (TIM4_PSCR)
- struct {
  _BITS ARR: 8
  auto-reload value
  } ARR

TIM4 8-bit auto-reload register (TIM4_ARR)

- struct {
  _BITS CEN: 1
  Counter enable.
  _BITS UDIS: 1
  Update disable.
  _BITS URS: 1
  Update request source.
  _BITS OPM: 1
  One-pulse mode.
  _BITS __pad0__: 3
  _BITS ARPE: 1
  Auto-reload preload enable.
  } CR1

TIM4 Control register 1 (TIM4_CR1)

- struct {
  _BITS __pad0__: 4
  _BITS MMS: 3
  Master mode selection.
  _BITS __pad1__: 1
  } CR2

TIM4 Control register 2 (TIM4_CR2)

- struct {
  _BITS SMS: 3
  Clock/trigger/slave mode selection.
  _BITS __pad0__: 1
  _BITS TS: 3
  Trigger selection.
  _BITS MSM: 1
  Master/slave mode.
  } SMCR

TIM4 Slave mode control register (TIM4_SMCR)

- struct {
  _BITS UIE: 1
  Update interrupt enable.
  _BITS __pad0__: 5
  _BITS TIE: 1
  Trigger interrupt enable.
  _BITS __pad1__: 1
  } IER

TIM4 Interrupt enable (TIM4_IER)

- struct {
  _BITS UIF: 1
  Update interrupt flag.
  _BITS __pad0__: 5
  _BITS TIF: 1
  Trigger interrupt flag.
  _BITS __pad1__: 1
  } SR1

TIM4 Status register (TIM4_SR1)

- struct {
  _BITS UG: 1

Update generation.
_BITS __pad0__: 5
_BITS TG: 1
Trigger generation.
_BITS __pad1__: 1
} EGR

TIM4 Event Generation (TIM4_EGR)

- struct {
_BITS CNT: 8
8-bit counter
} CNTR

TIM4 8-bit counter register (TIM4_CNTR)

- struct {
_BITS PSC: 4
clock prescaler
_BITS __pad0__: 4
} PSCR

TIM4 clock prescaler (TIM4_PSCR)

- struct {
_BITS ARR: 8
auto-reload value
} ARR

TIM4 8-bit auto-reload register (TIM4_ARR)

6.22.1 Detailed Description

struct for controlling 8-Bit Timer 4 (TIM4)

Definition at line 3819 of file STM8AF_STM8S.h.

6.22.2 Field Documentation

6.22.2.1 __pad0__

```
_BITS __pad0__
```

Definition at line 3827 of file STM8AF_STM8S.h.

6.22.2.2 __pad1__

```
_BITS __pad1__
```

Definition at line 2199 of file STM8L10x.h.

6.22.2.3 ARPE

_BITS ARPE

Auto-reload preload enable.

Definition at line 3828 of file STM8AF_STM8S.h.

6.22.2.4 ARR [1/3]

struct { ... } ARR

TIM4 8-bit auto-reload register (TIM4_ARR)

6.22.2.5 ARR [2/3]

_BITS ARR

auto-reload value

Definition at line 3873 of file STM8AF_STM8S.h.

6.22.2.6 ARR [3/3]

struct { ... } ARR

TIM4 8-bit auto-reload register (TIM4_ARR)

6.22.2.7 CEN

_BITS CEN

Counter enable.

Definition at line 3823 of file STM8AF_STM8S.h.

6.22.2.8 CNT

_BITS CNT

8-bit counter

Definition at line 3861 of file STM8AF_STM8S.h.

6.22.2.9 CNTR [1/2]

struct { ... } CNTR

TIM4 8-bit counter register (TIM4_CNTR)

6.22.2.10 CNTR [2/2]

struct { ... } CNTR

TIM4 8-bit counter register (TIM4_CNTR)

6.22.2.11 CR

struct { ... } CR

TIM4 Control register (TIM4_CR)

6.22.2.12 CR1

struct { ... } CR1

TIM4 Control register 1 (TIM4_CR1)

6.22.2.13 CR2

struct { ... } CR2

TIM4 Control register 2 (TIM4_CR2)

6.22.2.14 EGR [1/2]

```
struct { ...  } EGR
```

TIM4 Event Generation (TIM4_EGR)

6.22.2.15 EGR [2/2]

```
struct { ...  } EGR
```

TIM4 Event Generation (TIM4_EGR)

6.22.2.16 IER [1/2]

```
struct { ...  } IER
```

TIM4 Interrupt enable (TIM4_IER)

6.22.2.17 IER [2/2]

```
struct { ...  } IER
```

Reserved registers on selected devices (2B)

TIM4 Interrupt enable (TIM4_IER)

6.22.2.18 MMS

```
_BITS MMS
```

Master mode selection.

Definition at line 2198 of file STM8L10x.h.

6.22.2.19 MSM

```
_BITS MSM
```

Master/slave mode.

Definition at line 2208 of file STM8L10x.h.

6.22.2.20 OPM

_BITS OPM

One-pulse mode.

Definition at line 3826 of file STM8AF_STM8S.h.

6.22.2.21 PSC

_BITS PSC

clock prescaler

Definition at line 3866 of file STM8AF_STM8S.h.

6.22.2.22 PSCR [1/2]

struct { ... } PSCR

TIM4 clock prescaler (TIM4_PSCR)

6.22.2.23 PSCR [2/2]

struct { ... } PSCR

TIM4 clock prescaler (TIM4_PSCR)

6.22.2.24 SMCR

struct { ... } SMCR

TIM4 Slave mode control register (TIM4_SMCR)

6.22.2.25 SMS

_BITS SMS

Clock/trigger/slave mode selection.

Definition at line 2205 of file STM8L10x.h.

6.22.2.26 SR

```
struct { ...  } SR
```

TIM4 Status register (TIM4_SR)

6.22.2.27 SR1

```
struct { ...  } SR1
```

TIM4 Status register (TIM4_SR1)

6.22.2.28 TG

```
_BITS TG
```

Trigger generation.

Definition at line 2234 of file STM8L10x.h.

6.22.2.29 TIE

```
_BITS TIE
```

Trigger interrupt enable.

Definition at line 2216 of file STM8L10x.h.

6.22.2.30 TIF

```
_BITS TIF
```

Trigger interrupt flag.

Definition at line 2225 of file STM8L10x.h.

6.22.2.31 TS

_BITS TS

Trigger selection.

Definition at line 2207 of file STM8L10x.h.

6.22.2.32 UDIS

_BITS UDIS

Update disable.

Definition at line 3824 of file STM8AF_STM8S.h.

6.22.2.33 UG

_BITS UG

Update generation.

Definition at line 3854 of file STM8AF_STM8S.h.

6.22.2.34 UIE

_BITS UIE

Update interrupt enable.

Definition at line 3840 of file STM8AF_STM8S.h.

6.22.2.35 UIF

_BITS UIF

Update interrupt flag.

Definition at line 3847 of file STM8AF_STM8S.h.

6.22.2.36 URS

_BITS URS

Update request source.

Definition at line 3825 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following files:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h
- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.23 TIM5_t Struct Reference

struct for controlling 16-Bit Timer 5 (TIM5)

`#include <STM8AF_STM8S.h>`

Data Fields

- struct {
  _BITS CEN: 1
  Counter enable.
  _BITS UDIS: 1
  Update disable.
  _BITS URS: 1
  Update request source.
  _BITS OPM: 1
  One-pulse mode.
  _BITS __pad0__: 3
  _BITS ARPE: 1
  Auto-reload preload enable.
  } CR1

  TIM5 Control register 1 (TIM5_CR1)
- struct {
  _BITS CCPC: 1
  Capture/compare preloaded control.
  _BITS __pad0__: 1
  _BITS COMS: 1
  Capture/compare control update selection.
  _BITS __pad1__: 1
  _BITS MMS: 3
  Master mode selection.
  _BITS __pad2__: 1
  } CR2

  TIM5 Control register 2 (TIM5_CR2)

- struct {
  _BITS SMS: 3
  Clock/trigger/slave mode selection.
  _BITS __pad0__: 1
  _BITS TS: 3
  Trigger selection.
  _BITS MSM: 1
  Master/slave mode.
  } SMCR

Slave mode control register (TIM5_SMCR)

- struct {
  _BITS UIE: 1
  Update interrupt enable.
  _BITS CC1IE: 1
  Capture/compare 1 interrupt enable.
  _BITS CC2IE: 1
  Capture/compare 2 interrupt enable.
  _BITS CC3IE: 1
  Capture/compare 3 interrupt enable.
  _BITS __pad0__: 2
  _BITS TIE: 1
  Trigger interrupt enable.
  _BITS __pad1__: 1
  } IER

TIM5 Interrupt enable register (TIM5_IER)

- struct {
  _BITS UIF: 1
  Update interrupt flag.
  _BITS CC1IF: 1
  Capture/compare 1 interrupt flag.
  _BITS CC2IF: 1
  Capture/compare 2 interrupt flag.
  _BITS CC3IF: 1
  Capture/compare 3 interrupt flag.
  _BITS __pad0__: 2
  _BITS TIF: 1
  Trigger interrupt flag.
  _BITS __pad1__: 1
  } SR1

TIM5 Status register 1 (TIM5_SR1)

- struct {
  _BITS __pad0__: 1
  _BITS CC1OF: 1
  Capture/compare 1 overcapture flag.
  _BITS CC2OF: 1
  Capture/compare 2 overcapture flag.
  _BITS CC3OF: 1
  Capture/compare 3 overcapture flag.
  _BITS __pad1__: 4
  } SR2

TIM5 Status register 2 (TIM5_SR2)

- struct {
  _BITS UG: 1

Update generation.
_BITS CC1G: 1
Capture/compare 1 generation.
_BITS CC2G: 1
Capture/compare 2 generation.
_BITS CC3G: 1
Capture/compare 3 generation.
_BITS __pad0__: 2
_BITS TG: 1
Trigger generation.
_BITS __pad1__: 1
} EGR

TIM5 Event generation register (TIM5_EGR)

- union {
  struct {
  _BITS CC1S: 2
  Compare 1 selection.
  _BITS __pad0__: 1
  _BITS OC1PE: 1
  Output compare 1 preload enable.
  _BITS OC1M: 3
  Output compare 1 mode.
  _BITS __pad1__: 1
  } OUT
  bitwise access to register (output mode)
  struct {
  _BITS CC1S: 2
  Capture 1 selection.
  _BITS IC1PSC: 2
  Input capture 1 prescaler.
  _BITS IC1F: 4
  Input capture 1 filter.
  } IN
  bitwise access to register (input mode)
  } CCMR1

TIM5 Capture/compare mode register 1 (TIM5_CCMR1)

- union {
  struct {
  _BITS CC2S: 2
  Capture/compare 2 selection.
  _BITS __pad0__: 1
  _BITS OC2PE: 1
  Output compare 2 preload enable.
  _BITS OC2M: 3
  Output compare 2 mode.
  _BITS __pad1__: 1
  } OUT
  bitwise access to register (output mode)
  struct {
  _BITS CC2S: 2
  Capture/compare 2 selection.
  _BITS IC2PSC: 2
  Input capture 2 prescaler.
  _BITS IC2F: 4
  Input capture 2 filter.
  } IN

bitwise access to register (input mode)
} CCMR2

TIM5 Capture/compare mode register 2 (TIM5_CCMR2)

- union {
  struct {
  _BITS CC3S: 2
  Capture/compare 3 selection.
  _BITS __pad0__: 1
  _BITS OC3PE: 1
  Output compare 3 preload enable.
  _BITS OC3M: 3
  Output compare 3 mode.
  _BITS __pad1__: 1
  } OUT
  bitwise access to register (output mode)
  struct {
  _BITS CC3S: 2
  Capture/compare 3 selection.
  _BITS IC3PSC: 2
  Input capture 3 prescaler.
  _BITS IC3F: 4
  Input capture 3 filter.
  } IN
  bitwise access to register (input mode)
  } CCMR3

TIM5 Capture/compare mode register 3 (TIM5_CCMR3)

- struct {
  _BITS CC1E: 1
  Capture/compare 1 output enable.
  _BITS CC1P: 1
  Capture/compare 1 output polarity.
  _BITS __pad0__: 2
  _BITS CC2E: 1
  Capture/compare 2 output enable.
  _BITS CC2P: 1
  Capture/compare 2 output polarity.
  _BITS __pad1__: 2
  } CCER1

TIM5 Capture/compare enable register 1 (TIM5_CCER1)

- struct {
  _BITS CC3E: 1
  Capture/compare 3 output enable.
  _BITS CC3P: 1
  Capture/compare 3 output polarity.
  _BITS __pad0__: 6
  } CCER2

TIM5 Capture/compare enable register 2 (TIM5_CCER2)

- struct {
  _BITS CNT: 8
  16-bit counter [15:8]
  } CNTRH

TIM5 16-bit counter high byte (TIM5_CNTRH)

- struct {
  _BITS CNT: 8
  16-bit counter [7:0]
  } CNTRL

TIM5 16-bit counter low byte (TIM5_CNTRL)

- struct {
  _BITS PSC: 4
  clock prescaler
  _BITS __pad0__: 4
  } PSCR

TIM5 prescaler (TIM5_PSCR)

- struct {
  _BITS ARR: 8
  16-bit auto-reload value [15:8]
  } ARRH

TIM5 16-bit auto-reload value high byte (TIM5_ARRH)

- struct {
  _BITS ARR: 8
  16-bit auto-reload value [7:0]
  } ARRL

TIM5 16-bit auto-reload value low byte (TIM5_ARRL)

- struct {
  _BITS CCR1: 8
  16-bit capture/compare value 1 [15:8]
  } CCR1H

TIM5 16-bit capture/compare value 1 high byte (TIM5_CCR1H)

- struct {
  _BITS CCR1: 8
  16-bit capture/compare value 1 [7:0]
  } CCR1L

TIM5 16-bit capture/compare value 1 low byte (TIM5_CCR1L)

- struct {
  _BITS CCR2: 8
  16-bit capture/compare value 2 [15:8]
  } CCR2H

TIM5 16-bit capture/compare value 2 high byte (TIM5_CCR2H)

- struct {
  _BITS CCR2: 8
  16-bit capture/compare value 2 [7:0]
  } CCR2L

TIM5 16-bit capture/compare value 2 low byte (TIM5_CCR2L)

- struct {
  _BITS CCR3: 8
  16-bit capture/compare value 3 [15:8]
  } CCR3H

TIM5 16-bit capture/compare value 3 high byte (TIM5_CCR3H)

- struct {
  _BITS CCR3: 8
  16-bit capture/compare value 3 [7:0]
  } CCR3L

  TIM5 16-bit capture/compare value 3 low byte (TIM5_CCR3L)

6.23.1 Detailed Description

struct for controlling 16-Bit Timer 5 (TIM5)

Definition at line 3944 of file STM8AF_STM8S.h.

6.23.2 Field Documentation

6.23.2.1 __pad0__

_BITS __pad0__

Definition at line 3952 of file STM8AF_STM8S.h.

6.23.2.2 __pad1__

_BITS __pad1__

Definition at line 3962 of file STM8AF_STM8S.h.

6.23.2.3 __pad2__

_BITS __pad2__

Definition at line 3964 of file STM8AF_STM8S.h.

6.23.2.4 ARPE

_BITS ARPE

Auto-reload preload enable.

Definition at line 3953 of file STM8AF_STM8S.h.

6.23.2.5 ARR

_BITS ARR

16-bit auto-reload value [15:8]

16-bit auto-reload value [7:0]

Definition at line 4129 of file STM8AF_STM8S.h.

6.23.2.6 ARRH

struct { ... } ARRH

TIM5 16-bit auto-reload value high byte (TIM5_ARRH)

6.23.2.7 ARRL

struct { ... } ARRL

TIM5 16-bit auto-reload value low byte (TIM5_ARRL)

6.23.2.8 CC1E

_BITS CC1E

Capture/compare 1 output enable.

Definition at line 4091 of file STM8AF_STM8S.h.

6.23.2.9 CC1G

_BITS CC1G

Capture/compare 1 generation.

Definition at line 4014 of file STM8AF_STM8S.h.

6.23.2.10 CC1IE

_BITS CC1IE

Capture/compare 1 interrupt enable.

Definition at line 3980 of file STM8AF_STM8S.h.

6.23.2.11 CC1IF

_BITS CC1IF

Capture/compare 1 interrupt flag.

Definition at line 3992 of file STM8AF_STM8S.h.

6.23.2.12 CC1OF

_BITS CC1OF

Capture/compare 1 overcapture flag.

Definition at line 4004 of file STM8AF_STM8S.h.

6.23.2.13 CC1P

_BITS CC1P

Capture/compare 1 output polarity.

Definition at line 4092 of file STM8AF_STM8S.h.

6.23.2.14 CC1S

_BITS CC1S

Compare 1 selection.

Capture 1 selection.

Definition at line 4028 of file STM8AF_STM8S.h.

6.23.2.15 CC2E

_BITS CC2E

Capture/compare 2 output enable.

Definition at line 4094 of file STM8AF_STM8S.h.

6.23.2.16 CC2G

_BITS CC2G

Capture/compare 2 generation.

Definition at line 4015 of file STM8AF_STM8S.h.

6.23.2.17 CC2IE

_BITS CC2IE

Capture/compare 2 interrupt enable.

Definition at line 3981 of file STM8AF_STM8S.h.

6.23.2.18 CC2IF

_BITS CC2IF

Capture/compare 2 interrupt flag.

Definition at line 3993 of file STM8AF_STM8S.h.

6.23.2.19 CC2OF

_BITS CC2OF

Capture/compare 2 overcapture flag.

Definition at line 4005 of file STM8AF_STM8S.h.

6.23.2.20 CC2P

_BITS CC2P

Capture/compare 2 output polarity.

Definition at line 4095 of file STM8AF_STM8S.h.

6.23.2.21 CC2S

_BITS CC2S

Capture/compare 2 selection.

Definition at line 4050 of file STM8AF_STM8S.h.

6.23.2.22 CC3E

_BITS CC3E

Capture/compare 3 output enable.

Definition at line 4102 of file STM8AF_STM8S.h.

6.23.2.23 CC3G

_BITS CC3G

Capture/compare 3 generation.

Definition at line 4016 of file STM8AF_STM8S.h.

6.23.2.24 CC3IE

_BITS CC3IE

Capture/compare 3 interrupt enable.

Definition at line 3982 of file STM8AF_STM8S.h.

6.23.2.25 CC3IF

_BITS CC3IF

Capture/compare 3 interrupt flag.

Definition at line 3994 of file STM8AF_STM8S.h.

6.23.2.26 CC3OF

_BITS CC3OF

Capture/compare 3 overcapture flag.

Definition at line 4006 of file STM8AF_STM8S.h.

6.23.2.27 CC3P

_BITS CC3P

Capture/compare 3 output polarity.

Definition at line 4103 of file STM8AF_STM8S.h.

6.23.2.28 CC3S

_BITS CC3S

Capture/compare 3 selection.

Definition at line 4072 of file STM8AF_STM8S.h.

6.23.2.29 CCER1

struct { ... } CCER1

TIM5 Capture/compare enable register 1 (TIM5_CCER1)

6.23.2.30 CCER2

```
struct { ... } CCER2
```

TIM5 Capture/compare enable register 2 (TIM5_CCER2)

6.23.2.31 CCMR1

```
union { ... } CCMR1
```

TIM5 Capture/compare mode register 1 (TIM5_CCMR1)

6.23.2.32 CCMR2

```
union { ... } CCMR2
```

TIM5 Capture/compare mode register 2 (TIM5_CCMR2)

6.23.2.33 CCMR3

```
union { ... } CCMR3
```

TIM5 Capture/compare mode register 3 (TIM5_CCMR3)

6.23.2.34 CCPC

```
_BITS CCPC
```

Capture/compare preloaded control.

Definition at line 3959 of file STM8AF_STM8S.h.

6.23.2.35 CCR1

```
_BITS CCR1
```

16-bit capture/compare value 1 [15:8]

16-bit capture/compare value 1 [7:0]

Definition at line 4141 of file STM8AF_STM8S.h.

6.23.2.36 CCR1H

```
struct { ...  } CCR1H
```

TIM5 16-bit capture/compare value 1 high byte (TIM5_CCR1H)

6.23.2.37 CCR1L

```
struct { ...  } CCR1L
```

TIM5 16-bit capture/compare value 1 low byte (TIM5_CCR1L)

6.23.2.38 CCR2

```
_BITS CCR2
```

16-bit capture/compare value 2 [15:8]

16-bit capture/compare value 2 [7:0]

Definition at line 4153 of file STM8AF_STM8S.h.

6.23.2.39 CCR2H

```
struct { ...  } CCR2H
```

TIM5 16-bit capture/compare value 2 high byte (TIM5_CCR2H)

6.23.2.40 CCR2L

```
struct { ...  } CCR2L
```

TIM5 16-bit capture/compare value 2 low byte (TIM5_CCR2L)

6.23.2.41 CCR3

_BITS CCR3

16-bit capture/compare value 3 [15:8]

16-bit capture/compare value 3 [7:0]

Definition at line 4165 of file STM8AF_STM8S.h.

6.23.2.42 CCR3H

struct { ... } CCR3H

TIM5 16-bit capture/compare value 3 high byte (TIM5_CCR3H)

6.23.2.43 CCR3L

struct { ... } CCR3L

TIM5 16-bit capture/compare value 3 low byte (TIM5_CCR3L)

6.23.2.44 CEN

_BITS CEN

Counter enable.

Definition at line 3948 of file STM8AF_STM8S.h.

6.23.2.45 CNT

_BITS CNT

16-bit counter [15:8]

16-bit counter [7:0]

Definition at line 4110 of file STM8AF_STM8S.h.

6.23.2.46 CNTRH

```
struct { ...  } CNTRH
```

TIM5 16-bit counter high byte (TIM5_CNTRH)

6.23.2.47 CNTRL

```
struct { ...  } CNTRL
```

TIM5 16-bit counter low byte (TIM5_CNTRL)

6.23.2.48 COMS

```
_BITS COMS
```

Capture/compare control update selection.

Definition at line 3961 of file STM8AF_STM8S.h.

6.23.2.49 CR1

```
struct { ...  } CR1
```

TIM5 Control register 1 (TIM5_CR1)

6.23.2.50 CR2

```
struct { ...  } CR2
```

TIM5 Control register 2 (TIM5_CR2)

6.23.2.51 EGR

```
struct { ...  } EGR
```

TIM5 Event generation register (TIM5_EGR)

6.23.2.52 IC1F

_BITS IC1F

Input capture 1 filter.

Definition at line 4039 of file STM8AF_STM8S.h.

6.23.2.53 IC1PSC

_BITS IC1PSC

Input capture 1 prescaler.

Definition at line 4038 of file STM8AF_STM8S.h.

6.23.2.54 IC2F

_BITS IC2F

Input capture 2 filter.

Definition at line 4061 of file STM8AF_STM8S.h.

6.23.2.55 IC2PSC

_BITS IC2PSC

Input capture 2 prescaler.

Definition at line 4060 of file STM8AF_STM8S.h.

6.23.2.56 IC3F

_BITS IC3F

Input capture 3 filter.

Definition at line 4083 of file STM8AF_STM8S.h.

6.23.2.57 IC3PSC

_BITS IC3PSC

Input capture 3 prescaler.

Definition at line 4082 of file STM8AF_STM8S.h.

6.23.2.58 IER

struct { ... } IER

TIM5 Interrupt enable register (TIM5_IER)

6.23.2.59 IN [1/3]

struct { ... } IN

bitwise access to register (input mode)

6.23.2.60 IN [2/3]

struct { ... } IN

bitwise access to register (input mode)

6.23.2.61 IN [3/3]

struct { ... } IN

bitwise access to register (input mode)

6.23.2.62 MMS

_BITS MMS

Master mode selection.

Definition at line 3963 of file STM8AF_STM8S.h.

6.23.2.63 MSM

_BITS MSM

Master/slave mode.

Definition at line 3973 of file STM8AF_STM8S.h.

6.23.2.64 OC1M

_BITS OC1M

Output compare 1 mode.

Definition at line 4031 of file STM8AF_STM8S.h.

6.23.2.65 OC1PE

_BITS OC1PE

Output compare 1 preload enable.

Definition at line 4030 of file STM8AF_STM8S.h.

6.23.2.66 OC2M

_BITS OC2M

Output compare 2 mode.

Definition at line 4053 of file STM8AF_STM8S.h.

6.23.2.67 OC2PE

_BITS OC2PE

Output compare 2 preload enable.

Definition at line 4052 of file STM8AF_STM8S.h.

6.23.2.68 OC3M

_BITS OC3M

Output compare 3 mode.

Definition at line 4075 of file STM8AF_STM8S.h.

6.23.2.69 OC3PE

_BITS OC3PE

Output compare 3 preload enable.

Definition at line 4074 of file STM8AF_STM8S.h.

6.23.2.70 OPM

_BITS OPM

One-pulse mode.

Definition at line 3951 of file STM8AF_STM8S.h.

6.23.2.71 OUT [1/3]

struct { ... } OUT

bitwise access to register (output mode)

6.23.2.72 OUT [2/3]

struct { ... } OUT

bitwise access to register (output mode)

6.23.2.73 OUT [3/3]

```
struct { ...  } OUT
```

bitwise access to register (output mode)

6.23.2.74 PSC

```
_BITS PSC
```

clock prescaler

Definition at line 4122 of file STM8AF_STM8S.h.

6.23.2.75 PSCR

```
struct { ...  } PSCR
```

TIM5 prescaler (TIM5_PSCR)

6.23.2.76 SMCR

```
struct { ...  } SMCR
```

Slave mode control register (TIM5_SMCR)

6.23.2.77 SMS

```
_BITS SMS
```

Clock/trigger/slave mode selection.

Definition at line 3970 of file STM8AF_STM8S.h.

6.23.2.78 SR1

```
struct { ...  } SR1
```

TIM5 Status register 1 (TIM5_SR1)

6.23.2.79 SR2

```
struct { ...  } SR2
```

TIM5 Status register 2 (TIM5_SR2)

6.23.2.80 TG

```
_BITS TG
```

Trigger generation.

Definition at line 4018 of file STM8AF_STM8S.h.

6.23.2.81 TIE

```
_BITS TIE
```

Trigger interrupt enable.

Definition at line 3984 of file STM8AF_STM8S.h.

6.23.2.82 TIF

```
_BITS TIF
```

Trigger interrupt flag.

Definition at line 3996 of file STM8AF_STM8S.h.

6.23.2.83 TS

```
_BITS TS
```

Trigger selection.

Definition at line 3972 of file STM8AF_STM8S.h.

6.23.2.84 UDIS

_BITS UDIS

Update disable.

Definition at line 3949 of file STM8AF_STM8S.h.

6.23.2.85 UG

_BITS UG

Update generation.

Definition at line 4013 of file STM8AF_STM8S.h.

6.23.2.86 UIE

_BITS UIE

Update interrupt enable.

Definition at line 3979 of file STM8AF_STM8S.h.

6.23.2.87 UIF

_BITS UIF

Update interrupt flag.

Definition at line 3991 of file STM8AF_STM8S.h.

6.23.2.88 URS

_BITS URS

Update request source.

Definition at line 3950 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

6.24 TIM6_t Struct Reference

struct for controlling 8-Bit Timer 6 (TIM6)

`#include <STM8AF_STM8S.h>`

Data Fields

- struct {
  _BITS CEN: 1
  Counter enable.
  _BITS UDIS: 1
  Update disable.
  _BITS URS: 1
  Update request source.
  _BITS OPM: 1
  One-pulse mode.
  _BITS __pad0__: 3
  _BITS ARPE: 1
  Auto-reload preload enable.
  } CR1

  TIM6 Control register (TIM6_CR1)
- struct {
  _BITS __pad0__: 4
  _BITS MMS: 3
  Master mode selection.
  _BITS __pad1__: 1
  } CR2

  TIM6 Control register (TIM6_CR2)
- struct {
  _BITS SMS: 3
  Clock/trigger/slave mode selection.
  _BITS __pad0__: 1
  _BITS TS: 3
  Trigger selection.
  _BITS MSM: 1
  Master/slave mode #define _TIM5_CR1_RESET_VALUE ((uint8_t) 0x00) ///< TIM5 control register 1 reset value.
  } SMCR

  Slave mode control register (TIM6_SMCR)
- struct {
  _BITS UIE: 1
  Update interrupt enable.
  _BITS __pad0__: 7
  } IER

  TIM6 Interrupt enable (TIM6_IER)
- struct {
  _BITS UIF: 1
  Update interrupt flag.
  _BITS __pad0__: 7
  } SR

TIM6 Status register (TIM6_SR)

- struct {
  _BITS UG: 1
  Update generation.
  _BITS __pad0__: 7
  } EGR

TIM6 Event Generation (TIM6_EGR)

- struct {
  _BITS CNT: 8
  8-bit counter
  } CNTR

TIM6 8-bit counter register (TIM6_CNTR)

- struct {
  _BITS PSC: 3
  clock prescaler
  _BITS __pad0__: 5
  } PSCR

TIM6 clock prescaler (TIM6_PSCR)

- struct {
  _BITS ARR: 8
  auto-reload value
  } ARR

TIM6 8-bit auto-reload register (TIM6_ARR)

6.24.1 Detailed Description

struct for controlling 8-Bit Timer 6 (TIM6)

Definition at line 4392 of file STM8AF_STM8S.h.

6.24.2 Field Documentation

6.24.2.1 __pad0__

```
_BITS __pad0__
```

Definition at line 4400 of file STM8AF_STM8S.h.

6.24.2.2 __pad1__

```
_BITS __pad1__
```

Definition at line 4409 of file STM8AF_STM8S.h.

6.24.2.3 ARPE

_BITS ARPE

Auto-reload preload enable.

Definition at line 4401 of file STM8AF_STM8S.h.

6.24.2.4 ARR [1/2]

_BITS ARR

auto-reload value

Definition at line 4458 of file STM8AF_STM8S.h.

6.24.2.5 ARR [2/2]

struct { ... } ARR

TIM6 8-bit auto-reload register (TIM6_ARR)

6.24.2.6 CEN

_BITS CEN

Counter enable.

Definition at line 4396 of file STM8AF_STM8S.h.

6.24.2.7 CNT

_BITS CNT

8-bit counter

Definition at line 4446 of file STM8AF_STM8S.h.

6.24.2.8 CNTR

```
struct { ... } CNTR
```

TIM6 8-bit counter register (TIM6_CNTR)

6.24.2.9 CR1

```
struct { ... } CR1
```

TIM6 Control register (TIM6_CR1)

6.24.2.10 CR2

```
struct { ... } CR2
```

TIM6 Control register (TIM6_CR2)

6.24.2.11 EGR

```
struct { ... } EGR
```

TIM6 Event Generation (TIM6_EGR)

6.24.2.12 IER

```
struct { ... } IER
```

TIM6 Interrupt enable (TIM6_IER)

6.24.2.13 MMS

```
_BITS MMS
```

Master mode selection.

Definition at line 4408 of file STM8AF_STM8S.h.

6.24.2.14 MSM

_BITS MSM

Master/slave mode #define _TIM5_CR1_RESET_VALUE ((uint8_t) 0x00) ///< TIM5 control register 1 reset value.

Definition at line 4418 of file STM8AF_STM8S.h.

6.24.2.15 OPM

_BITS OPM

One-pulse mode.

Definition at line 4399 of file STM8AF_STM8S.h.

6.24.2.16 PSC

_BITS PSC

clock prescaler

Definition at line 4451 of file STM8AF_STM8S.h.

6.24.2.17 PSCR

struct { ... } PSCR

TIM6 clock prescaler (TIM6_PSCR)

6.24.2.18 SMCR

struct { ... } SMCR

Slave mode control register (TIM6_SMCR)

6.24.2.19 SMS

_BITS SMS

Clock/trigger/slave mode selection.

Definition at line 4415 of file STM8AF_STM8S.h.

6.24.2.20 SR

struct { ... } SR

TIM6 Status register (TIM6_SR)

6.24.2.21 TS

_BITS TS

Trigger selection.

Definition at line 4417 of file STM8AF_STM8S.h.

6.24.2.22 UDIS

_BITS UDIS

Update disable.

Definition at line 4397 of file STM8AF_STM8S.h.

6.24.2.23 UG

_BITS UG

Update generation.

Definition at line 4439 of file STM8AF_STM8S.h.

6.24.2.24 UIE

_BITS UIE

Update interrupt enable.

Definition at line 4425 of file STM8AF_STM8S.h.

6.24.2.25 UIF

_BITS UIF

Update interrupt flag.

Definition at line 4432 of file STM8AF_STM8S.h.

6.24.2.26 URS

_BITS URS

Update request source.

Definition at line 4398 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

6.25 UART1_t Struct Reference

struct for controlling Universal Asynchronous Receiver Transmitter 1 (UART1)

#include <STM8AF_STM8S.h>

Data Fields

- struct {
  _BITS PE: 1
  Parity error.
  _BITS FE: 1
  Framing error.
  _BITS NF: 1
  Noise flag.
  _BITS OR: 1
  LIN Header Error (LIN Slave mode) / Overrun error.
  _BITS IDLE: 1
  IDLE line detected.
  _BITS RXNE: 1
  Read data register not empty.
  _BITS TC: 1
  Transmission complete.
  _BITS TXE: 1
  Transmit data register empty.
  } SR

  UART1 Status register (UART1_SR)
- struct {
  _BITS DATA: 8
  UART1 data.
  } DR

  UART1 data register (UART1_DR)
- struct {
  _BITS DIV_4_11: 8
  UART_DIV bits [11:4].
  } BRR1

  UART1 Baud rate register 1 (UART1_BRR1)
- struct {
  _BITS DIV_0_3: 4
  UART_DIV bits [3:0].
  _BITS DIV_12_15: 4
  UART_DIV bits [15:12].
  } BRR2

  UART1 Baud rate register 2 (UART1_BRR2)
- struct {
  _BITS PIEN: 1
  Parity interrupt enable.
  _BITS PS: 1
  Parity selection.
  _BITS PCEN: 1
  Parity control enable.
  _BITS WAKE: 1
  Wakeup method.
  _BITS M: 1
  word length
  _BITS UARTD: 1
  UART Disable (for low power consumption)
  _BITS T8: 1
  Transmit Data bit 8 (in 9-bit mode)

_BITS R8: 1
Receive Data bit 8 (in 9-bit mode)
} CR1

UART1 Control register 1 (UART1_CR1)

- struct {
  _BITS SBK: 1
  Send break.
  _BITS RWU: 1
  Receiver wakeup.
  _BITS REN: 1
  Receiver enable.
  _BITS TEN: 1
  Transmitter enable.
  _BITS ILIEN: 1
  IDLE Line interrupt enable.
  _BITS RIEN: 1
  Receiver interrupt enable.
  _BITS TCIEN: 1
  Transmission complete interrupt enable.
  _BITS TIEN: 1
  Transmitter interrupt enable.
  } CR2

UART1 Control register 2 (UART1_CR2)

- struct {
  _BITS LBCL: 1
  Last bit clock pulse.
  _BITS CPHA: 1
  Clock phase.
  _BITS CPOL: 1
  Clock polarity.
  _BITS CKEN: 1
  Clock enable.
  _BITS STOP: 2
  STOP bits.
  _BITS LINEN: 1
  LIN mode enable.
  _BITS __pad0__: 1
  } CR3

UART1 Control register 3 (UART1_CR3)

- struct {
  _BITS ADD: 4
  Address of the UART node.
  _BITS LBDF: 1
  LIN Break Detection Flag.
  _BITS LBDL: 1
  LIN Break Detection Length.
  _BITS LBDIEN: 1
  LIN Break Detection Interrupt Enable.
  _BITS __pad0__: 1
  } CR4

UART1 Control register 4 (UART1_CR4)

- struct {
  _BITS __pad0__: 1

_BITS IREN: 1
IrDA mode Enable.
_BITS IRLP: 1
IrDA Low Power.
_BITS HDSEL: 1
Half-Duplex Selection.
_BITS NACK: 1
Smartcard NACK enable.
_BITS SCEN: 1
Smartcard mode enable.
_BITS __pad1__: 2
} CR5

UART1 Control register 5 (UART1_CR5)

- struct {
_BITS GT: 8
UART1 guard time.
} GTR

UART1 guard time register (UART1_GTR)

- struct {
_BITS PSC: 8
UART1 prescaler.
} PSCR

UART1 prescaler register (UART1_PSCR)

6.25.1 Detailed Description

struct for controlling Universal Asynchronous Receiver Transmitter 1 (UART1)

Definition at line 1623 of file STM8AF_STM8S.h.

6.25.2 Field Documentation

6.25.2.1 __pad0__

_BITS __pad0__

Definition at line 1691 of file STM8AF_STM8S.h.

6.25.2.2 __pad1__

_BITS __pad1__

Definition at line 1713 of file STM8AF_STM8S.h.

6.25.2.3 ADD

_BITS ADD

Address of the UART node.

Definition at line 1697 of file STM8AF_STM8S.h.

6.25.2.4 BRR1

struct { ... } BRR1

UART1 Baud rate register 1 (UART1_BRR1)

6.25.2.5 BRR2

struct { ... } BRR2

UART1 Baud rate register 2 (UART1_BRR2)

6.25.2.6 CKEN

_BITS CKEN

Clock enable.

Definition at line 1688 of file STM8AF_STM8S.h.

6.25.2.7 CPHA

_BITS CPHA

Clock phase.

Definition at line 1686 of file STM8AF_STM8S.h.

6.25.2.8 CPOL

_BITS CPOL

Clock polarity.

Definition at line 1687 of file STM8AF_STM8S.h.

6.25.2.9 CR1

struct { ... } CR1

UART1 Control register 1 (UART1_CR1)

6.25.2.10 CR2

struct { ... } CR2

UART1 Control register 2 (UART1_CR2)

6.25.2.11 CR3

struct { ... } CR3

UART1 Control register 3 (UART1_CR3)

6.25.2.12 CR4

struct { ... } CR4

UART1 Control register 4 (UART1_CR4)

6.25.2.13 CR5

struct { ... } CR5

UART1 Control register 5 (UART1_CR5)

6.25.2.14 DATA

_BITS DATA

UART1 data.

Definition at line 1640 of file STM8AF_STM8S.h.

6.25.2.15 DIV_0_3

_BITS DIV_0_3

UART_DIV bits [3:0].

Definition at line 1652 of file STM8AF_STM8S.h.

6.25.2.16 DIV_12_15

_BITS DIV_12_15

UART_DIV bits [15:12].

Definition at line 1653 of file STM8AF_STM8S.h.

6.25.2.17 DIV_4_11

_BITS DIV_4_11

UART_DIV bits [11:4].

Definition at line 1646 of file STM8AF_STM8S.h.

6.25.2.18 DR

struct { ... } DR

UART1 data register (UART1_DR)

6.25.2.19 FE

_BITS FE

Framing error.

Definition at line 1628 of file STM8AF_STM8S.h.

6.25.2.20 GT

_BITS GT

UART1 guard time.

Definition at line 1719 of file STM8AF_STM8S.h.

6.25.2.21 GTR

struct { ... } GTR

UART1 guard time register (UART1_GTR)

6.25.2.22 HDSEL

_BITS HDSEL

Half-Duplex Selection.

Definition at line 1710 of file STM8AF_STM8S.h.

6.25.2.23 IDLE

_BITS IDLE

IDLE line detected.

Definition at line 1631 of file STM8AF_STM8S.h.

6.25.2.24 ILIEN

_BITS ILIEN

IDLE Line interrupt enable.

Definition at line 1676 of file STM8AF_STM8S.h.

6.25.2.25 IREN

_BITS IREN

IrDA mode Enable.

Definition at line 1708 of file STM8AF_STM8S.h.

6.25.2.26 IRLP

_BITS IRLP

IrDA Low Power.

Definition at line 1709 of file STM8AF_STM8S.h.

6.25.2.27 LBCL

_BITS LBCL

Last bit clock pulse.

Definition at line 1685 of file STM8AF_STM8S.h.

6.25.2.28 LBDF

_BITS LBDF

LIN Break Detection Flag.

Definition at line 1698 of file STM8AF_STM8S.h.

6.25.2.29 LBDIEN

_BITS LBDIEN

LIN Break Detection Interrupt Enable.

Definition at line 1700 of file STM8AF_STM8S.h.

6.25.2.30 LBDL

_BITS LBDL

LIN Break Detection Length.

Definition at line 1699 of file STM8AF_STM8S.h.

6.25.2.31 LINEN

_BITS LINEN

LIN mode enable.

Definition at line 1690 of file STM8AF_STM8S.h.

6.25.2.32 M

_BITS M

word length

Definition at line 1663 of file STM8AF_STM8S.h.

6.25.2.33 NACK

_BITS NACK

Smartcard NACK enable.

Definition at line 1711 of file STM8AF_STM8S.h.

6.25.2.34 NF

_BITS NF

Noise flag.

Definition at line 1629 of file STM8AF_STM8S.h.

6.25.2.35 OR

_BITS OR

LIN Header Error (LIN Slave mode) / Overrun error.

Definition at line 1630 of file STM8AF_STM8S.h.

6.25.2.36 PCEN

_BITS PCEN

Parity control enable.

Definition at line 1661 of file STM8AF_STM8S.h.

6.25.2.37 PE

_BITS PE

Parity error.

Definition at line 1627 of file STM8AF_STM8S.h.

6.25.2.38 PIEN

_BITS PIEN

Parity interrupt enable.

Definition at line 1659 of file STM8AF_STM8S.h.

6.25.2.39 PS

_BITS PS

Parity selection.

Definition at line 1660 of file STM8AF_STM8S.h.

6.25.2.40 PSC

_BITS PSC

UART1 prescaler.

Definition at line 1725 of file STM8AF_STM8S.h.

6.25.2.41 PSCR

struct { ... } PSCR

UART1 prescaler register (UART1_PSCR)

6.25.2.42 R8

_BITS R8

Receive Data bit 8 (in 9-bit mode)

Definition at line 1666 of file STM8AF_STM8S.h.

6.25.2.43 REN

_BITS REN

Receiver enable.

Definition at line 1674 of file STM8AF_STM8S.h.

6.25.2.44 RIEN

_BITS RIEN

Receiver interrupt enable.

Definition at line 1677 of file STM8AF_STM8S.h.

6.25.2.45 RWU

_BITS RWU

Receiver wakeup.

Definition at line 1673 of file STM8AF_STM8S.h.

6.25.2.46 RXNE

_BITS RXNE

Read data register not empty.

Definition at line 1632 of file STM8AF_STM8S.h.

6.25.2.47 SBK

_BITS SBK

Send break.

Definition at line 1672 of file STM8AF_STM8S.h.

6.25.2.48 SCEN

_BITS SCEN

Smartcard mode enable.

Definition at line 1712 of file STM8AF_STM8S.h.

6.25.2.49 SR

```
struct { ...  } SR
```

UART1 Status register (UART1_SR)

6.25.2.50 STOP

```
_BITS STOP
```

STOP bits.

Definition at line 1689 of file STM8AF_STM8S.h.

6.25.2.51 T8

```
_BITS T8
```

Transmit Data bit 8 (in 9-bit mode)

Definition at line 1665 of file STM8AF_STM8S.h.

6.25.2.52 TC

```
_BITS TC
```

Transmission complete.

Definition at line 1633 of file STM8AF_STM8S.h.

6.25.2.53 TCIEN

```
_BITS TCIEN
```

Transmission complete interrupt enable.

Definition at line 1678 of file STM8AF_STM8S.h.

6.25.2.54 TEN

_BITS TEN

Transmitter enable.

Definition at line 1675 of file STM8AF_STM8S.h.

6.25.2.55 TIEN

_BITS TIEN

Transmitter interrupt enable.

Definition at line 1679 of file STM8AF_STM8S.h.

6.25.2.56 TXE

_BITS TXE

Transmit data register empty.

Definition at line 1634 of file STM8AF_STM8S.h.

6.25.2.57 UARTD

_BITS UARTD

UART Disable (for low power consumption)

Definition at line 1664 of file STM8AF_STM8S.h.

6.25.2.58 WAKE

_BITS WAKE

Wakeup method.

Definition at line 1662 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

6.26 UART2_t Struct Reference

struct for controlling Universal Asynchronous Receiver Transmitter 2 (UART2)

```
#include <STM8AF_STM8S.h>
```

Data Fields

- struct {
  _BITS PE: 1
  Parity error.
  _BITS FE: 1
  Framing error.
  _BITS NF: 1
  Noise flag.
  _BITS OR: 1
  LIN Header Error (LIN slave mode) / Overrun error.
  _BITS IDLE: 1
  IDLE line detected.
  _BITS RXNE: 1
  Read data register not empty.
  _BITS TC: 1
  Transmission complete.
  _BITS TXE: 1
  Transmit data register empty.
  } SR

  UART2 Status register (UART2_SR)
- struct {
  _BITS DATA: 8
  UART2 data.
  } DR

  UART2 data register (UART2_DR)
- struct {
  _BITS DIV_4_11: 8
  UART2_BRR bits [11:4].
  } BRR1

  UART2 Baud rate register 1 (UART2_BRR1)
- struct {
  _BITS DIV_0_3: 4
  UART2_BRR bits [3:0].
  _BITS DIV_12_15: 4
  UART2_BRR bits [15:12].
  } BRR2

  UART2 Baud rate register 2 (UART2_BRR2)
- struct {
  _BITS PIEN: 1
  Parity interrupt enable.
  _BITS PS: 1
  Parity selection.
  _BITS PCEN: 1
  Parity control enable.

_BITS WAKE: 1
Wakeup method.
_BITS M: 1
word length
_BITS UARTD: 1
UART Disable (for low power consumption)
_BITS T8: 1
Transmit Data bit 8 (in 9-bit mode)
_BITS R8: 1
Receive Data bit 8 (in 9-bit mode)
} CR1

UART2 Control register 1 (UART2_CR1)

- struct {
_BITS SBK: 1
Send break.
_BITS RWU: 1
Receiver wakeup.
_BITS REN: 1
Receiver enable.
_BITS TEN: 1
Transmitter enable.
_BITS ILIEN: 1
IDLE Line interrupt enable.
_BITS RIEN: 1
Receiver interrupt enable.
_BITS TCIEN: 1
Transmission complete interrupt enable.
_BITS TIEN: 1
Transmitter interrupt enable.
} CR2

UART2 Control register 2 (UART2_CR2)

- struct {
_BITS LBCL: 1
Last bit clock pulse.
_BITS CPHA: 1
Clock phase.
_BITS CPOL: 1
Clock polarity.
_BITS CKEN: 1
Clock enable.
_BITS STOP: 2
STOP bits.
_BITS LINEN: 1
LIN mode enable.
_BITS __pad0__: 1
} CR3

UART2 Control register 3 (UART2_CR3)

- struct {
_BITS ADD: 4
Address of the UART node.
_BITS LBDF: 1
LIN Break Detection Flag.
_BITS LBDL: 1
LIN Break Detection Length.
_BITS LBDIEN: 1

LIN Break Detection Interrupt Enable.
_BITS __pad0__: 1
} CR4

UART2 Control register 4 (UART2_CR4)

- struct {
_BITS __pad0__: 1
_BITS IREN: 1
IrDA mode Enable.
_BITS IRLP: 1
IrDA Low Power.
_BITS __pad1__: 1
_BITS NACK: 1
Smartcard NACK enable.
_BITS SCEN: 1
Smartcard mode enable.
_BITS __pad2__: 2
} CR5

UART2 Control register 5 (UART2_CR5)

- struct {
_BITS LSF: 1
LIN Sync Field.
_BITS LHDF: 1
LIN Header Detection Flag.
_BITS LHDIEN: 1
LIN Header Detection Interrupt Enable.
_BITS __pad0__: 1
_BITS LASE: 1
LIN automatic resynchronisation enable.
_BITS LSLV: 1
LIN Slave Enable.
_BITS __pad1__: 1
_BITS LDUM: 1
LIN Divider Update Method.
} CR6

UART2 Control register 6 (UART2_CR6)

- struct {
_BITS GT: 8
UART2 guard time.
} GTR

UART2 guard time register (UART2_GTR)

- struct {
_BITS PSC: 8
UART1 prescaler.
} PSCR

UART2 prescaler register (UART2_PSCR)

6.26.1 Detailed Description

struct for controlling Universal Asynchronous Receiver Transmitter 2 (UART2)

Definition at line 1827 of file STM8AF_STM8S.h.

6.26.2 Field Documentation

6.26.2.1 __pad0__

_BITS __pad0__

Definition at line 1895 of file STM8AF_STM8S.h.

6.26.2.2 __pad1__

_BITS __pad1__

Definition at line 1914 of file STM8AF_STM8S.h.

6.26.2.3 __pad2__

_BITS __pad2__

Definition at line 1917 of file STM8AF_STM8S.h.

6.26.2.4 ADD

_BITS ADD

Address of the UART node.

Definition at line 1901 of file STM8AF_STM8S.h.

6.26.2.5 BRR1

struct { ... } BRR1

UART2 Baud rate register 1 (UART2_BRR1)

6.26.2.6 BRR2

```
struct { ...  } BRR2
```

UART2 Baud rate register 2 (UART2_BRR2)

6.26.2.7 CKEN

```
_BITS CKEN
```

Clock enable.

Definition at line 1892 of file STM8AF_STM8S.h.

6.26.2.8 CPHA

```
_BITS CPHA
```

Clock phase.

Definition at line 1890 of file STM8AF_STM8S.h.

6.26.2.9 CPOL

```
_BITS CPOL
```

Clock polarity.

Definition at line 1891 of file STM8AF_STM8S.h.

6.26.2.10 CR1

```
struct { ...  } CR1
```

UART2 Control register 1 (UART2_CR1)

6.26.2.11 CR2

```
struct { ...  } CR2
```

UART2 Control register 2 (UART2_CR2)

6.26.2.12 CR3

```
struct { ...  } CR3
```

UART2 Control register 3 (UART2_CR3)

6.26.2.13 CR4

```
struct { ...  } CR4
```

UART2 Control register 4 (UART2_CR4)

6.26.2.14 CR5

```
struct { ...  } CR5
```

UART2 Control register 5 (UART2_CR5)

6.26.2.15 CR6

```
struct { ...  } CR6
```

UART2 Control register 6 (UART2_CR6)

6.26.2.16 DATA

```
_BITS DATA
```

UART2 data.

Definition at line 1844 of file STM8AF_STM8S.h.

6.26.2.17 DIV_0_3

_BITS DIV_0_3

UART2_BRR bits [3:0].

Definition at line 1856 of file STM8AF_STM8S.h.

6.26.2.18 DIV_12_15

_BITS DIV_12_15

UART2_BRR bits [15:12].

Definition at line 1857 of file STM8AF_STM8S.h.

6.26.2.19 DIV_4_11

_BITS DIV_4_11

UART2_BRR bits [11:4].

Definition at line 1850 of file STM8AF_STM8S.h.

6.26.2.20 DR

struct { ... } DR

UART2 data register (UART2_DR)

6.26.2.21 FE

_BITS FE

Framing error.

Definition at line 1832 of file STM8AF_STM8S.h.

6.26.2.22 GT

_BITS GT

UART2 guard time.

Definition at line 1936 of file STM8AF_STM8S.h.

6.26.2.23 GTR

struct { ... } GTR

UART2 guard time register (UART2_GTR)

6.26.2.24 IDLE

_BITS IDLE

IDLE line detected.

Definition at line 1835 of file STM8AF_STM8S.h.

6.26.2.25 ILIEN

_BITS ILIEN

IDLE Line interrupt enable.

Definition at line 1880 of file STM8AF_STM8S.h.

6.26.2.26 IREN

_BITS IREN

IrDA mode Enable.

Definition at line 1912 of file STM8AF_STM8S.h.

6.26.2.27 IRLP

_BITS IRLP

IrDA Low Power.

Definition at line 1913 of file STM8AF_STM8S.h.

6.26.2.28 LASE

_BITS LASE

LIN automatic resynchronisation enable.

Definition at line 1927 of file STM8AF_STM8S.h.

6.26.2.29 LBCL

_BITS LBCL

Last bit clock pulse.

Definition at line 1889 of file STM8AF_STM8S.h.

6.26.2.30 LBDF

_BITS LBDF

LIN Break Detection Flag.

Definition at line 1902 of file STM8AF_STM8S.h.

6.26.2.31 LBDIEN

_BITS LBDIEN

LIN Break Detection Interrupt Enable.

Definition at line 1904 of file STM8AF_STM8S.h.

6.26.2.32 LBDL

_BITS LBDL

LIN Break Detection Length.

Definition at line 1903 of file STM8AF_STM8S.h.

6.26.2.33 LDUM

_BITS LDUM

LIN Divider Update Method.

Definition at line 1930 of file STM8AF_STM8S.h.

6.26.2.34 LHDF

_BITS LHDF

LIN Header Detection Flag.

Definition at line 1924 of file STM8AF_STM8S.h.

6.26.2.35 LHDIEN

_BITS LHDIEN

LIN Header Detection Interrupt Enable.

Definition at line 1925 of file STM8AF_STM8S.h.

6.26.2.36 LINEN

_BITS LINEN

LIN mode enable.

Definition at line 1894 of file STM8AF_STM8S.h.

6.26.2.37 LSF

_BITS LSF

LIN Sync Field.

Definition at line 1923 of file STM8AF_STM8S.h.

6.26.2.38 LSLV

_BITS LSLV

LIN Slave Enable.

Definition at line 1928 of file STM8AF_STM8S.h.

6.26.2.39 M

_BITS M

word length

Definition at line 1867 of file STM8AF_STM8S.h.

6.26.2.40 NACK

_BITS NACK

Smartcard NACK enable.

Definition at line 1915 of file STM8AF_STM8S.h.

6.26.2.41 NF

_BITS NF

Noise flag.

Definition at line 1833 of file STM8AF_STM8S.h.

6.26.2.42 OR

_BITS OR

LIN Header Error (LIN slave mode) / Overrun error.

Definition at line 1834 of file STM8AF_STM8S.h.

6.26.2.43 PCEN

_BITS PCEN

Parity control enable.

Definition at line 1865 of file STM8AF_STM8S.h.

6.26.2.44 PE

_BITS PE

Parity error.

Definition at line 1831 of file STM8AF_STM8S.h.

6.26.2.45 PIEN

_BITS PIEN

Parity interrupt enable.

Definition at line 1863 of file STM8AF_STM8S.h.

6.26.2.46 PS

_BITS PS

Parity selection.

Definition at line 1864 of file STM8AF_STM8S.h.

6.26.2.47 PSC

_BITS PSC

UART1 prescaler.

Definition at line 1942 of file STM8AF_STM8S.h.

6.26.2.48 PSCR

struct { ... } PSCR

UART2 prescaler register (UART2_PSCR)

6.26.2.49 R8

_BITS R8

Receive Data bit 8 (in 9-bit mode)

Definition at line 1870 of file STM8AF_STM8S.h.

6.26.2.50 REN

_BITS REN

Receiver enable.

Definition at line 1878 of file STM8AF_STM8S.h.

6.26.2.51 RIEN

_BITS RIEN

Receiver interrupt enable.

Definition at line 1881 of file STM8AF_STM8S.h.

6.26.2.52 RWU

_BITS RWU

Receiver wakeup.

Definition at line 1877 of file STM8AF_STM8S.h.

6.26.2.53 RXNE

_BITS RXNE

Read data register not empty.

Definition at line 1836 of file STM8AF_STM8S.h.

6.26.2.54 SBK

_BITS SBK

Send break.

Definition at line 1876 of file STM8AF_STM8S.h.

6.26.2.55 SCEN

_BITS SCEN

Smartcard mode enable.

Definition at line 1916 of file STM8AF_STM8S.h.

6.26.2.56 SR

struct { ... } SR

UART2 Status register (UART2_SR)

6.26.2.57 STOP

_BITS STOP

STOP bits.

Definition at line 1893 of file STM8AF_STM8S.h.

6.26.2.58 T8

_BITS T8

Transmit Data bit 8 (in 9-bit mode)

Definition at line 1869 of file STM8AF_STM8S.h.

6.26.2.59 TC

_BITS TC

Transmission complete.

Definition at line 1837 of file STM8AF_STM8S.h.

6.26.2.60 TCIEN

_BITS TCIEN

Transmission complete interrupt enable.

Definition at line 1882 of file STM8AF_STM8S.h.

6.26.2.61 TEN

_BITS TEN

Transmitter enable.

Definition at line 1879 of file STM8AF_STM8S.h.

6.26.2.62 TIEN

_BITS TIEN

Transmitter interrupt enable.

Definition at line 1883 of file STM8AF_STM8S.h.

6.26.2.63 TXE

_BITS TXE

Transmit data register empty.

Definition at line 1838 of file STM8AF_STM8S.h.

6.26.2.64 UARTD

_BITS UARTD

UART Disable (for low power consumption)

Definition at line 1868 of file STM8AF_STM8S.h.

6.26.2.65 WAKE

_BITS WAKE

Wakeup method.

Definition at line 1866 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

6.27 UART3_t Struct Reference

struct for controlling Universal Asynchronous Receiver Transmitter 3 (UART3)

#include <STM8AF_STM8S.h>

Data Fields

- struct {
  _BITS PE: 1
  Parity error.
  _BITS FE: 1
  Framing error.
  _BITS NF: 1
  Noise flag.
  _BITS OR: 1
  LIN Header Error (LIN slave mode) / Overrun error.
  _BITS IDLE: 1
  IDLE line detected.
  _BITS RXNE: 1
  Read data register not empty.
  _BITS TC: 1
  Transmission complete.
  _BITS TXE: 1
  Transmit data register empty.
  } SR

  UART3 Status register (UART3_SR)
- struct {
  _BITS DATA: 8
  UART3 data.
  } DR

  UART3 data register (UART3_DR)
- struct {
  _BITS DIV_4_11: 8
  UART_DIV bits [11:4].
  } BRR1

  UART3 Baud rate register 1 (UART3_BRR1)
- struct {
  _BITS DIV_0_3: 4
  UART_DIV bits [3:0].
  _BITS DIV_12_15: 4
  UART_DIV bits [15:12].
  } BRR2

  UART3 Baud rate register 2 (UART3_BRR2)
- struct {
  _BITS PIEN: 1
  Parity interrupt enable.
  _BITS PS: 1
  Parity selection.
  _BITS PCEN: 1
  Parity control enable.
  _BITS WAKE: 1
  Wakeup method.
  _BITS M: 1
  word length
  _BITS UARTD: 1
  UART Disable (for low power consumption)
  _BITS T8: 1
  Transmit Data bit 8 (in 9-bit mode)

_BITS R8: 1
Receive Data bit 8 (in 9-bit mode)
} CR1

UART3 Control register 1 (UART3_CR1)

- struct {
_BITS SBK: 1
Send break.
_BITS RWU: 1
Receiver wakeup.
_BITS REN: 1
Receiver enable.
_BITS TEN: 1
Transmitter enable.
_BITS ILIEN: 1
IDLE Line interrupt enable.
_BITS RIEN: 1
Receiver interrupt enable.
_BITS TCIEN: 1
Transmission complete interrupt enable.
_BITS TIEN: 1
Transmitter interrupt enable.
} CR2

UART3 Control register 2 (UART3_CR2)

- struct {
_BITS __pad0__: 4
_BITS STOP: 2
STOP bits.
_BITS LINEN: 1
LIN mode enable.
_BITS __pad1__: 1
} CR3

UART3 Control register 3 (UART3_CR3)

- struct {
_BITS ADD: 4
Address of the UART node.
_BITS LBDF: 1
LIN Break Detection Flag.
_BITS LBDL: 1
LIN Break Detection Length.
_BITS LBDIEN: 1
LIN Break Detection Interrupt Enable.
_BITS __pad0__: 1
} CR4

UART3 Control register 4 (UART3_CR4)

- uint8_t res [1]

Reserved register (1B)

- struct {
_BITS LSF: 1
LIN Sync Field.
_BITS LHDF: 1
LIN Header Detection Flag.
_BITS LHDIEN: 1
LIN Header Detection Interrupt Enable.

_BITS __pad0__: 1
_BITS LASE: 1
LIN automatic resynchronisation enable.
_BITS LSLV: 1
LIN Slave Enable.
_BITS __pad1__: 1
_BITS LDUM: 1
LIN Divider Update Method.
} CR6

UART3 Control register 6 (UART3_CR6)

6.27.1 Detailed Description

struct for controlling Universal Asynchronous Receiver Transmitter 3 (UART3)

Definition at line 2056 of file STM8AF_STM8S.h.

6.27.2 Field Documentation

6.27.2.1 __pad0__

_BITS __pad0__

Definition at line 2118 of file STM8AF_STM8S.h.

6.27.2.2 __pad1__

_BITS __pad1__

Definition at line 2121 of file STM8AF_STM8S.h.

6.27.2.3 ADD

_BITS ADD

Address of the UART node.

Definition at line 2127 of file STM8AF_STM8S.h.

6.27.2.4 BRR1

```
struct { ...  } BRR1
```

UART3 Baud rate register 1 (UART3_BRR1)

6.27.2.5 BRR2

```
struct { ...  } BRR2
```

UART3 Baud rate register 2 (UART3_BRR2)

6.27.2.6 CR1

```
struct { ...  } CR1
```

UART3 Control register 1 (UART3_CR1)

6.27.2.7 CR2

```
struct { ...  } CR2
```

UART3 Control register 2 (UART3_CR2)

6.27.2.8 CR3

```
struct { ...  } CR3
```

UART3 Control register 3 (UART3_CR3)

6.27.2.9 CR4

```
struct { ...  } CR4
```

UART3 Control register 4 (UART3_CR4)

6.27.2.10 CR6

```
struct { ...  } CR6
```

UART3 Control register 6 (UART3_CR6)

6.27.2.11 DATA

```
_BITS DATA
```

UART3 data.

Definition at line 2073 of file STM8AF_STM8S.h.

6.27.2.12 DIV_0_3

```
_BITS DIV_0_3
```

UART_DIV bits [3:0].

Definition at line 2085 of file STM8AF_STM8S.h.

6.27.2.13 DIV_12_15

```
_BITS DIV_12_15
```

UART_DIV bits [15:12].

Definition at line 2086 of file STM8AF_STM8S.h.

6.27.2.14 DIV_4_11

```
_BITS DIV_4_11
```

UART_DIV bits [11:4].

Definition at line 2079 of file STM8AF_STM8S.h.

6.27.2.15 DR

```
struct { ...  } DR
```

UART3 data register (UART3_DR)

6.27.2.16 FE

```
_BITS FE
```

Framing error.

Definition at line 2061 of file STM8AF_STM8S.h.

6.27.2.17 IDLE

```
_BITS IDLE
```

IDLE line detected.

Definition at line 2064 of file STM8AF_STM8S.h.

6.27.2.18 ILIEN

```
_BITS ILIEN
```

IDLE Line interrupt enable.

Definition at line 2109 of file STM8AF_STM8S.h.

6.27.2.19 LASE

```
_BITS LASE
```

LIN automatic resynchronisation enable.

Definition at line 2145 of file STM8AF_STM8S.h.

6.27.2.20 LBDF

_BITS LBDF

LIN Break Detection Flag.

Definition at line 2128 of file STM8AF_STM8S.h.

6.27.2.21 LBDIEN

_BITS LBDIEN

LIN Break Detection Interrupt Enable.

Definition at line 2130 of file STM8AF_STM8S.h.

6.27.2.22 LBDL

_BITS LBDL

LIN Break Detection Length.

Definition at line 2129 of file STM8AF_STM8S.h.

6.27.2.23 LDUM

_BITS LDUM

LIN Divider Update Method.

Definition at line 2148 of file STM8AF_STM8S.h.

6.27.2.24 LHDF

_BITS LHDF

LIN Header Detection Flag.

Definition at line 2142 of file STM8AF_STM8S.h.

6.27.2.25 LHDIEN

_BITS LHDIEN

LIN Header Detection Interrupt Enable.

Definition at line 2143 of file STM8AF_STM8S.h.

6.27.2.26 LINEN

_BITS LINEN

LIN mode enable.

Definition at line 2120 of file STM8AF_STM8S.h.

6.27.2.27 LSF

_BITS LSF

LIN Sync Field.

Definition at line 2141 of file STM8AF_STM8S.h.

6.27.2.28 LSLV

_BITS LSLV

LIN Slave Enable.

Definition at line 2146 of file STM8AF_STM8S.h.

6.27.2.29 M

_BITS M

word length

Definition at line 2096 of file STM8AF_STM8S.h.

6.27.2.30 NF

_BITS NF

Noise flag.

Definition at line 2062 of file STM8AF_STM8S.h.

6.27.2.31 OR

_BITS OR

LIN Header Error (LIN slave mode) / Overrun error.

Definition at line 2063 of file STM8AF_STM8S.h.

6.27.2.32 PCEN

_BITS PCEN

Parity control enable.

Definition at line 2094 of file STM8AF_STM8S.h.

6.27.2.33 PE

_BITS PE

Parity error.

Definition at line 2060 of file STM8AF_STM8S.h.

6.27.2.34 PIEN

_BITS PIEN

Parity interrupt enable.

Definition at line 2092 of file STM8AF_STM8S.h.

6.27.2.35 PS

_BITS PS

Parity selection.

Definition at line 2093 of file STM8AF_STM8S.h.

6.27.2.36 R8

_BITS R8

Receive Data bit 8 (in 9-bit mode)

Definition at line 2099 of file STM8AF_STM8S.h.

6.27.2.37 REN

_BITS REN

Receiver enable.

Definition at line 2107 of file STM8AF_STM8S.h.

6.27.2.38 res

uint8_t res[1]

Reserved register (1B)

Definition at line 2136 of file STM8AF_STM8S.h.

6.27.2.39 RIEN

_BITS RIEN

Receiver interrupt enable.

Definition at line 2110 of file STM8AF_STM8S.h.

6.27.2.40 RWU

_BITS RWU

Receiver wakeup.

Definition at line 2106 of file STM8AF_STM8S.h.

6.27.2.41 RXNE

_BITS RXNE

Read data register not empty.

Definition at line 2065 of file STM8AF_STM8S.h.

6.27.2.42 SBK

_BITS SBK

Send break.

Definition at line 2105 of file STM8AF_STM8S.h.

6.27.2.43 SR

struct { ... } SR

UART3 Status register (UART3_SR)

6.27.2.44 STOP

_BITS STOP

STOP bits.

Definition at line 2119 of file STM8AF_STM8S.h.

6.27.2.45 T8

_BITS T8

Transmit Data bit 8 (in 9-bit mode)

Definition at line 2098 of file STM8AF_STM8S.h.

6.27.2.46 TC

_BITS TC

Transmission complete.

Definition at line 2066 of file STM8AF_STM8S.h.

6.27.2.47 TCIEN

_BITS TCIEN

Transmission complete interrupt enable.

Definition at line 2111 of file STM8AF_STM8S.h.

6.27.2.48 TEN

_BITS TEN

Transmitter enable.

Definition at line 2108 of file STM8AF_STM8S.h.

6.27.2.49 TIEN

_BITS TIEN

Transmitter interrupt enable.

Definition at line 2112 of file STM8AF_STM8S.h.

6.27.2.50 TXE

_BITS TXE

Transmit data register empty.

Definition at line 2067 of file STM8AF_STM8S.h.

6.27.2.51 UARTD

_BITS UARTD

UART Disable (for low power consumption)

Definition at line 2097 of file STM8AF_STM8S.h.

6.27.2.52 WAKE

_BITS WAKE

Wakeup method.

Definition at line 2095 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

6.28 UART4_t Struct Reference

struct for controlling Universal Asynchronous Receiver Transmitter 4 (UART4)

#include <STM8AF_STM8S.h>

Data Fields

- struct {
  _BITS PE: 1
  Parity error.
  _BITS FE: 1
  Framing error.
  _BITS NF: 1
  Noise flag.
  _BITS OR: 1
  LIN Header Error (LIN slave mode) / Overrun error.
  _BITS IDLE: 1
  IDLE line detected.
  _BITS RXNE: 1
  Read data register not empty.
  _BITS TC: 1
  Transmission complete.
  _BITS TXE: 1
  Transmit data register empty.
  } SR

  UART4 Status register (UART4_SR)
- struct {
  _BITS DATA: 8
  UART4 data.
  } DR

  UART4 data register (UART4_DR)
- struct {
  _BITS DIV_4_11: 8
  UART_DIV bits [11:4].
  } BRR1

  UART4 Baud rate register 1 (UART4_BRR1)
- struct {
  _BITS DIV_0_3: 4
  UART_DIV bits [3:0].
  _BITS DIV_12_15: 4
  UART_DIV bits [15:12].
  } BRR2

  UART4 Baud rate register 2 (UART4_BRR2)
- struct {
  _BITS PIEN: 1
  Parity interrupt enable.
  _BITS PS: 1
  Parity selection.
  _BITS PCEN: 1
  Parity control enable.
  _BITS WAKE: 1
  Wakeup method.
  _BITS M: 1
  word length
  _BITS UARTD: 1
  UART Disable (for low power consumption)
  _BITS T8: 1
  Transmit Data bit 8 (in 9-bit mode)

_BITS R8: 1
Receive Data bit 8 (in 9-bit mode)
} CR1

UART4 Control register 1 (UART4_CR1)

- struct {
_BITS SBK: 1
Send break.
_BITS RWU: 1
Receiver wakeup.
_BITS REN: 1
Receiver enable.
_BITS TEN: 1
Transmitter enable.
_BITS ILIEN: 1
IDLE Line interrupt enable.
_BITS RIEN: 1
Receiver interrupt enable.
_BITS TCIEN: 1
Transmission complete interrupt enable.
_BITS TIEN: 1
Transmitter interrupt enable.
} CR2

UART4 Control register 2 (UART4_CR2)

- struct {
_BITS LBCL: 1
Last bit clock pulse.
_BITS CPHA: 1
Clock phase.
_BITS CPOL: 1
Clock polarity.
_BITS CKEN: 1
Clock enable.
_BITS STOP: 2
STOP bits.
_BITS LINEN: 1
LIN mode enable.
_BITS __pad0__: 1
} CR3

UART4 Control register 3 (UART4_CR3)

- struct {
_BITS ADD: 4
Address of the UART node.
_BITS LBDF: 1
LIN Break Detection Flag.
_BITS LBDL: 1
LIN Break Detection Length.
_BITS LBDIEN: 1
LIN Break Detection Interrupt Enable.
_BITS __pad0__: 1
} CR4

UART4 Control register 4 (UART4_CR4)

- struct {
_BITS __pad0__: 1

_BITS IREN: 1
IrDA mode Enable.
_BITS IRLP: 1
IrDA Low Power.
_BITS HDSEL: 1
Half-Duplex Selection.
_BITS NACK: 1
Smartcard NACK enable.
_BITS SCEN: 1
Smartcard mode enable.
_BITS __pad1__: 2
} CR5

UART4 Control register 5 (UART4_CR5)

- struct {
_BITS LSF: 1
LIN Sync Field.
_BITS LHDF: 1
LIN Header Detection Flag.
_BITS LHDIEN: 1
LIN Header Detection Interrupt Enable.
_BITS __pad0__: 1
_BITS LASE: 1
LIN automatic resynchronisation enable.
_BITS LSLV: 1
LIN Slave Enable.
_BITS __pad1__: 1
_BITS LDUM: 1
LIN Divider Update Method.
} CR6

UART4 Control register 6 (UART4_CR6)

- struct {
_BITS GT: 8
UART4 guard time.
} GTR

UART4 guard time register (UART4_GTR)

- struct {
_BITS PSC: 8
UART4 prescaler.
} PSCR

UART4 prescaler register (UART4_PSCR)

6.28.1 Detailed Description

struct for controlling Universal Asynchronous Receiver Transmitter 4 (UART4)

Definition at line 2245 of file STM8AF_STM8S.h.

6.28.2 Field Documentation

6.28.2.1 __pad0__

_BITS __pad0__

Definition at line 2313 of file STM8AF_STM8S.h.

6.28.2.2 __pad1__

_BITS __pad1__

Definition at line 2335 of file STM8AF_STM8S.h.

6.28.2.3 ADD

_BITS ADD

Address of the UART node.

Definition at line 2319 of file STM8AF_STM8S.h.

6.28.2.4 BRR1

struct { ... } BRR1

UART4 Baud rate register 1 (UART4_BRR1)

6.28.2.5 BRR2

struct { ... } BRR2

UART4 Baud rate register 2 (UART4_BRR2)

6.28.2.6 CKEN

_BITS CKEN

Clock enable.

Definition at line 2310 of file STM8AF_STM8S.h.

6.28.2.7 CPHA

_BITS CPHA

Clock phase.

Definition at line 2308 of file STM8AF_STM8S.h.

6.28.2.8 CPOL

_BITS CPOL

Clock polarity.

Definition at line 2309 of file STM8AF_STM8S.h.

6.28.2.9 CR1

struct { ... } CR1

UART4 Control register 1 (UART4_CR1)

6.28.2.10 CR2

struct { ... } CR2

UART4 Control register 2 (UART4_CR2)

6.28.2.11 CR3

struct { ... } CR3

UART4 Control register 3 (UART4_CR3)

6.28.2.12 CR4

struct { ... } CR4

UART4 Control register 4 (UART4_CR4)

6.28.2.13 CR5

```
struct { ...  } CR5
```

UART4 Control register 5 (UART4_CR5)

6.28.2.14 CR6

```
struct { ...  } CR6
```

UART4 Control register 6 (UART4_CR6)

6.28.2.15 DATA

```
_BITS DATA
```

UART4 data.

Definition at line 2262 of file STM8AF_STM8S.h.

6.28.2.16 DIV_0_3

```
_BITS DIV_0_3
```

UART_DIV bits [3:0].

Definition at line 2274 of file STM8AF_STM8S.h.

6.28.2.17 DIV_12_15

```
_BITS DIV_12_15
```

UART_DIV bits [15:12].

Definition at line 2275 of file STM8AF_STM8S.h.

6.28.2.18 DIV_4_11

_BITS DIV_4_11

UART_DIV bits [11:4].

Definition at line 2268 of file STM8AF_STM8S.h.

6.28.2.19 DR

struct { ... } DR

UART4 data register (UART4_DR)

6.28.2.20 FE

_BITS FE

Framing error.

Definition at line 2250 of file STM8AF_STM8S.h.

6.28.2.21 GT

_BITS GT

UART4 guard time.

Definition at line 2354 of file STM8AF_STM8S.h.

6.28.2.22 GTR

struct { ... } GTR

UART4 guard time register (UART4_GTR)

6.28.2.23 HDSEL

_BITS HDSEL

Half-Duplex Selection.

Definition at line 2332 of file STM8AF_STM8S.h.

6.28.2.24 IDLE

_BITS IDLE

IDLE line detected.

Definition at line 2253 of file STM8AF_STM8S.h.

6.28.2.25 ILIEN

_BITS ILIEN

IDLE Line interrupt enable.

Definition at line 2298 of file STM8AF_STM8S.h.

6.28.2.26 IREN

_BITS IREN

IrDA mode Enable.

Definition at line 2330 of file STM8AF_STM8S.h.

6.28.2.27 IRLP

_BITS IRLP

IrDA Low Power.

Definition at line 2331 of file STM8AF_STM8S.h.

6.28.2.28 LASE

_BITS LASE

LIN automatic resynchronisation enable.

Definition at line 2345 of file STM8AF_STM8S.h.

6.28.2.29 LBCL

_BITS LBCL

Last bit clock pulse.

Definition at line 2307 of file STM8AF_STM8S.h.

6.28.2.30 LBDF

_BITS LBDF

LIN Break Detection Flag.

Definition at line 2320 of file STM8AF_STM8S.h.

6.28.2.31 LBDIEN

_BITS LBDIEN

LIN Break Detection Interrupt Enable.

Definition at line 2322 of file STM8AF_STM8S.h.

6.28.2.32 LBDL

_BITS LBDL

LIN Break Detection Length.

Definition at line 2321 of file STM8AF_STM8S.h.

6.28.2.33 LDUM

_BITS LDUM

LIN Divider Update Method.

Definition at line 2348 of file STM8AF_STM8S.h.

6.28.2.34 LHDF

_BITS LHDF

LIN Header Detection Flag.

Definition at line 2342 of file STM8AF_STM8S.h.

6.28.2.35 LHDIEN

_BITS LHDIEN

LIN Header Detection Interrupt Enable.

Definition at line 2343 of file STM8AF_STM8S.h.

6.28.2.36 LINEN

_BITS LINEN

LIN mode enable.

Definition at line 2312 of file STM8AF_STM8S.h.

6.28.2.37 LSF

_BITS LSF

LIN Sync Field.

Definition at line 2341 of file STM8AF_STM8S.h.

6.28.2.38 LSLV

_BITS LSLV

LIN Slave Enable.

Definition at line 2346 of file STM8AF_STM8S.h.

6.28.2.39 M

_BITS M

word length

Definition at line 2285 of file STM8AF_STM8S.h.

6.28.2.40 NACK

_BITS NACK

Smartcard NACK enable.

Definition at line 2333 of file STM8AF_STM8S.h.

6.28.2.41 NF

_BITS NF

Noise flag.

Definition at line 2251 of file STM8AF_STM8S.h.

6.28.2.42 OR

_BITS OR

LIN Header Error (LIN slave mode) / Overrun error.

Definition at line 2252 of file STM8AF_STM8S.h.

6.28.2.43 PCEN

_BITS PCEN

Parity control enable.

Definition at line 2283 of file STM8AF_STM8S.h.

6.28.2.44 PE

_BITS PE

Parity error.

Definition at line 2249 of file STM8AF_STM8S.h.

6.28.2.45 PIEN

_BITS PIEN

Parity interrupt enable.

Definition at line 2281 of file STM8AF_STM8S.h.

6.28.2.46 PS

_BITS PS

Parity selection.

Definition at line 2282 of file STM8AF_STM8S.h.

6.28.2.47 PSC

_BITS PSC

UART4 prescaler.

Definition at line 2360 of file STM8AF_STM8S.h.

6.28.2.48 PSCR

```
struct { ...  } PSCR
```

UART4 prescaler register (UART4_PSCR)

6.28.2.49 R8

```
_BITS R8
```

Receive Data bit 8 (in 9-bit mode)

Definition at line 2288 of file STM8AF_STM8S.h.

6.28.2.50 REN

```
_BITS REN
```

Receiver enable.

Definition at line 2296 of file STM8AF_STM8S.h.

6.28.2.51 RIEN

```
_BITS RIEN
```

Receiver interrupt enable.

Definition at line 2299 of file STM8AF_STM8S.h.

6.28.2.52 RWU

```
_BITS RWU
```

Receiver wakeup.

Definition at line 2295 of file STM8AF_STM8S.h.

6.28.2.53 RXNE

_BITS RXNE

Read data register not empty.

Definition at line 2254 of file STM8AF_STM8S.h.

6.28.2.54 SBK

_BITS SBK

Send break.

Definition at line 2294 of file STM8AF_STM8S.h.

6.28.2.55 SCEN

_BITS SCEN

Smartcard mode enable.

Definition at line 2334 of file STM8AF_STM8S.h.

6.28.2.56 SR

struct { ... } SR

UART4 Status register (UART4_SR)

6.28.2.57 STOP

_BITS STOP

STOP bits.

Definition at line 2311 of file STM8AF_STM8S.h.

6.28.2.58 T8

_BITS T8

Transmit Data bit 8 (in 9-bit mode)

Definition at line 2287 of file STM8AF_STM8S.h.

6.28.2.59 TC

_BITS TC

Transmission complete.

Definition at line 2255 of file STM8AF_STM8S.h.

6.28.2.60 TCIEN

_BITS TCIEN

Transmission complete interrupt enable.

Definition at line 2300 of file STM8AF_STM8S.h.

6.28.2.61 TEN

_BITS TEN

Transmitter enable.

Definition at line 2297 of file STM8AF_STM8S.h.

6.28.2.62 TIEN

_BITS TIEN

Transmitter interrupt enable.

Definition at line 2301 of file STM8AF_STM8S.h.

6.28.2.63 TXE

_BITS TXE

Transmit data register empty.

Definition at line 2256 of file STM8AF_STM8S.h.

6.28.2.64 UARTD

_BITS UARTD

UART Disable (for low power consumption)

Definition at line 2286 of file STM8AF_STM8S.h.

6.28.2.65 WAKE

_BITS WAKE

Wakeup method.

Definition at line 2284 of file STM8AF_STM8S.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

6.29 USART_t Struct Reference

struct for controlling Universal Asynchronous Receiver Transmitter (USART)

```
#include <STM8L10x.h>
```

Data Fields

- struct {
  _BITS PE: 1
  Parity error.
  _BITS FE: 1
  Framing error.
  _BITS NF: 1
  Noise flag.
  _BITS OR: 1
  LIN Header Error (LIN Slave mode) / Overrun error.
  _BITS IDLE: 1
  IDLE line detected.
  _BITS RXNE: 1
  Read data register not empty.
  _BITS TC: 1
  Transmission complete.
  _BITS TXE: 1
  Transmit data register empty.
  } SR

  USART Status register (USART_SR)
- struct {
  _BITS DATA: 8
  USART data.
  } DR

  USART data register (USART_DR)
- struct {
  _BITS DIV_4_11: 8
  UART_DIV bits [11:4].
  } BRR1

  USART Baud rate register 1 (USART_BRR1)
- struct {
  _BITS DIV_0_3: 4
  UART_DIV bits [3:0].
  _BITS DIV_12_15: 4
  UART_DIV bits [15:12].
  } BRR2

  USART Baud rate register 2 (USART_BRR2)
- struct {
  _BITS PIEN: 1
  Parity interrupt enable.
  _BITS PS: 1
  Parity selection.
  _BITS PCEN: 1
  Parity control enable.
  _BITS WAKE: 1
  Wakeup method.
  _BITS M: 1
  word length
  _BITS UARTD: 1
  UART Disable (for low power consumption)
  _BITS T8: 1
  Transmit Data bit 8 (in 9-bit mode)

_BITS R8: 1
Receive Data bit 8 (in 9-bit mode)
} CR1

USART Control register 1 (USART_CR1)

- struct {
_BITS SBK: 1
Send break.
_BITS RWU: 1
Receiver wakeup.
_BITS REN: 1
Receiver enable.
_BITS TEN: 1
Transmitter enable.
_BITS ILIEN: 1
IDLE Line interrupt enable.
_BITS RIEN: 1
Receiver interrupt enable.
_BITS TCIEN: 1
Transmission complete interrupt enable.
_BITS TIEN: 1
Transmitter interrupt enable.
} CR2

USART Control register 2 (USART_CR2)

- struct {
_BITS LBCL: 1
Last bit clock pulse.
_BITS CPHA: 1
Clock phase.
_BITS CPOL: 1
Clock polarity.
_BITS CKEN: 1
Clock enable.
_BITS STOP: 2
STOP bits.
_BITS __pad0__: 2
} CR3

USART Control register 3 (USART_CR3)

- struct {
_BITS ADD: 4
Address of the UART node.
_BITS __pad0__: 4
} CR4

USART Control register 4 (USART_CR4)

6.29.1 Detailed Description

struct for controlling Universal Asynchronous Receiver Transmitter (USART)

Definition at line 1332 of file STM8L10x.h.

6.29.2 Field Documentation

6.29.2.1 __pad0__

_BITS __pad0__

Definition at line 1399 of file STM8L10x.h.

6.29.2.2 ADD

_BITS ADD

Address of the UART node.

Definition at line 1405 of file STM8L10x.h.

6.29.2.3 BRR1

struct { ... } BRR1

USART Baud rate register 1 (USART_BRR1)

6.29.2.4 BRR2

struct { ... } BRR2

USART Baud rate register 2 (USART_BRR2)

6.29.2.5 CKEN

_BITS CKEN

Clock enable.

Definition at line 1397 of file STM8L10x.h.

6.29.2.6 CPHA

```
_BITS CPHA
```

Clock phase.

Definition at line 1395 of file STM8L10x.h.

6.29.2.7 CPOL

```
_BITS CPOL
```

Clock polarity.

Definition at line 1396 of file STM8L10x.h.

6.29.2.8 CR1

```
struct { ...  } CR1
```

USART Control register 1 (USART_CR1)

6.29.2.9 CR2

```
struct { ...  } CR2
```

USART Control register 2 (USART_CR2)

6.29.2.10 CR3

```
struct { ...  } CR3
```

USART Control register 3 (USART_CR3)

6.29.2.11 CR4

```
struct { ...  } CR4
```

USART Control register 4 (USART_CR4)

6.29.2.12 DATA

_BITS DATA

USART data.

Definition at line 1349 of file STM8L10x.h.

6.29.2.13 DIV_0_3

_BITS DIV_0_3

UART_DIV bits [3:0].

Definition at line 1361 of file STM8L10x.h.

6.29.2.14 DIV_12_15

_BITS DIV_12_15

UART_DIV bits [15:12].

Definition at line 1362 of file STM8L10x.h.

6.29.2.15 DIV_4_11

_BITS DIV_4_11

UART_DIV bits [11:4].

Definition at line 1355 of file STM8L10x.h.

6.29.2.16 DR

struct { ... } DR

USART data register (USART_DR)

6.29.2.17 FE

_BITS FE

Framing error.

Definition at line 1337 of file STM8L10x.h.

6.29.2.18 IDLE

_BITS IDLE

IDLE line detected.

Definition at line 1340 of file STM8L10x.h.

6.29.2.19 ILIEN

_BITS ILIEN

IDLE Line interrupt enable.

Definition at line 1385 of file STM8L10x.h.

6.29.2.20 LBCL

_BITS LBCL

Last bit clock pulse.

Definition at line 1394 of file STM8L10x.h.

6.29.2.21 M

_BITS M

word length

Definition at line 1372 of file STM8L10x.h.

6.29.2.22 NF

_BITS NF

Noise flag.

Definition at line 1338 of file STM8L10x.h.

6.29.2.23 OR

_BITS OR

LIN Header Error (LIN Slave mode) / Overrun error.

Definition at line 1339 of file STM8L10x.h.

6.29.2.24 PCEN

_BITS PCEN

Parity control enable.

Definition at line 1370 of file STM8L10x.h.

6.29.2.25 PE

_BITS PE

Parity error.

Definition at line 1336 of file STM8L10x.h.

6.29.2.26 PIEN

_BITS PIEN

Parity interrupt enable.

Definition at line 1368 of file STM8L10x.h.

6.29.2.27 PS

_BITS PS

Parity selection.

Definition at line 1369 of file STM8L10x.h.

6.29.2.28 R8

_BITS R8

Receive Data bit 8 (in 9-bit mode)

Definition at line 1375 of file STM8L10x.h.

6.29.2.29 REN

_BITS REN

Receiver enable.

Definition at line 1383 of file STM8L10x.h.

6.29.2.30 RIEN

_BITS RIEN

Receiver interrupt enable.

Definition at line 1386 of file STM8L10x.h.

6.29.2.31 RWU

_BITS RWU

Receiver wakeup.

Definition at line 1382 of file STM8L10x.h.

6.29.2.32 RXNE

_BITS RXNE

Read data register not empty.

Definition at line 1341 of file STM8L10x.h.

6.29.2.33 SBK

_BITS SBK

Send break.

Definition at line 1381 of file STM8L10x.h.

6.29.2.34 SR

struct { ... } SR

USART Status register (USART_SR)

6.29.2.35 STOP

_BITS STOP

STOP bits.

Definition at line 1398 of file STM8L10x.h.

6.29.2.36 T8

_BITS T8

Transmit Data bit 8 (in 9-bit mode)

Definition at line 1374 of file STM8L10x.h.

6.29.2.37 TC

_BITS TC

Transmission complete.

Definition at line 1342 of file STM8L10x.h.

6.29.2.38 TCIEN

_BITS TCIEN

Transmission complete interrupt enable.

Definition at line 1387 of file STM8L10x.h.

6.29.2.39 TEN

_BITS TEN

Transmitter enable.

Definition at line 1384 of file STM8L10x.h.

6.29.2.40 TIEN

_BITS TIEN

Transmitter interrupt enable.

Definition at line 1388 of file STM8L10x.h.

6.29.2.41 TXE

_BITS TXE

Transmit data register empty.

Definition at line 1343 of file STM8L10x.h.

6.29.2.42 UARTD

_BITS UARTD

UART Disable (for low power consumption)

Definition at line 1373 of file STM8L10x.h.

6.29.2.43 WAKE

_BITS WAKE

Wakeup method.

Definition at line 1371 of file STM8L10x.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.30 WFE_t Struct Reference

struct to configure interrupt sources as external interrupts or wake events (WFE)

```
#include <STM8L10x.h>
```

Data Fields

- struct {
  _BITS TIM2_EV0: 1
  TIM2 update, trigger or break event.
  _BITS TIM2_EV1: 1
  TIM2 capture or compare event.
  _BITS __pad0__: 2
  _BITS EXTI_EV0: 1
  Interrupt on pin 0 of all ports event.
  _BITS EXTI_EV1: 1
  Interrupt on pin 1 of all ports event.
  _BITS EXTI_EV2: 1
  Interrupt on pin 2 of all ports event.
  _BITS EXTI_EV3: 1
  Interrupt on pin 3 of all ports event.
  } CR1

  WFE control register 1 (WFE_CR1)

- struct {
  _BITS EXTI_EV4: 1
  Interrupt on pin 4 of all ports event.
  _BITS EXTI_EV5: 1
  Interrupt on pin 5 of all ports event.
  _BITS EXTI_EV6: 1
  Interrupt on pin 6 of all ports event.
  _BITS EXTI_EV7: 1
  Interrupt on pin 7 of all ports event.
  _BITS EXTI_EVB: 1
  Interrupt on port B event.
  _BITS EXTI_EVD: 1
  Interrupt on port D event.
  _BITS __pad0__: 2
  } CR2

  WFE control register 2 (WFE_CR2)

6.30.1 Detailed Description

struct to configure interrupt sources as external interrupts or wake events (WFE)

Definition at line 1489 of file STM8L10x.h.

6.30.2 Field Documentation

6.30.2.1 __pad0__

```
_BITS __pad0__
```

Definition at line 1495 of file STM8L10x.h.

6.30.2.2 CR1

```
struct { ... } CR1
```

WFE control register 1 (WFE_CR1)

6.30.2.3 CR2

```
struct { ... } CR2
```

WFE control register 2 (WFE_CR2)

6.30.2.4 EXTI_EV0

_BITS EXTI_EV0

Interrupt on pin 0 of all ports event.

Definition at line 1496 of file STM8L10x.h.

6.30.2.5 EXTI_EV1

_BITS EXTI_EV1

Interrupt on pin 1 of all ports event.

Definition at line 1497 of file STM8L10x.h.

6.30.2.6 EXTI_EV2

_BITS EXTI_EV2

Interrupt on pin 2 of all ports event.

Definition at line 1498 of file STM8L10x.h.

6.30.2.7 EXTI_EV3

_BITS EXTI_EV3

Interrupt on pin 3 of all ports event.

Definition at line 1499 of file STM8L10x.h.

6.30.2.8 EXTI_EV4

_BITS EXTI_EV4

Interrupt on pin 4 of all ports event.

Definition at line 1505 of file STM8L10x.h.

6.30.2.9 EXTI_EV5

_BITS EXTI_EV5

Interrupt on pin 5 of all ports event.

Definition at line 1506 of file STM8L10x.h.

6.30.2.10 EXTI_EV6

_BITS EXTI_EV6

Interrupt on pin 6 of all ports event.

Definition at line 1507 of file STM8L10x.h.

6.30.2.11 EXTI_EV7

_BITS EXTI_EV7

Interrupt on pin 7 of all ports event.

Definition at line 1508 of file STM8L10x.h.

6.30.2.12 EXTI_EVB

_BITS EXTI_EVB

Interrupt on port B event.

Definition at line 1509 of file STM8L10x.h.

6.30.2.13 EXTI_EVD

_BITS EXTI_EVD

Interrupt on port D event.

Definition at line 1510 of file STM8L10x.h.

6.30.2.14 TIM2_EV0

_BITS TIM2_EV0

TIM2 update, trigger or break event.

Definition at line 1493 of file STM8L10x.h.

6.30.2.15 TIM2_EV1

_BITS TIM2_EV1

TIM2 capture or compare event.

Definition at line 1494 of file STM8L10x.h.

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h

6.31 WWDG_t Struct Reference

struct for access to Window Watchdog registers (WWDG)

`#include <STM8AF_STM8S.h>`

Data Fields

- struct {
  _BITS T: 7
  7-bit WWDG counter
  _BITS WDGA: 1
  WWDG activation bit (n/a if WWDG enabled by option byte)
  } CR

  WWDG Control register (WWDG_CR)
- struct {
  _BITS W: 7
  7-bit window value
  _BITS __pad0__: 1
  } WR

  WWDR Window register (WWDG_WR)

6.31.1 Detailed Description

struct for access to Window Watchdog registers (WWDG)

Definition at line 988 of file STM8AF_STM8S.h.

6.31.2 Field Documentation

6.31.2.1 __pad0__

_BITS __pad0__

Definition at line 1000 of file STM8AF_STM8S.h.

6.31.2.2 CR

struct { ... } CR

WWDG Control register (WWDG_CR)

6.31.2.3 T

_BITS T

7-bit WWDG counter

Definition at line 992 of file STM8AF_STM8S.h.

6.31.2.4 W

_BITS W

7-bit window value

Definition at line 999 of file STM8AF_STM8S.h.

6.31.2.5 WDGA

_BITS WDGA

WWDG activation bit (n/a if WWDG enabled by option byte)

Definition at line 993 of file STM8AF_STM8S.h.

6.31.2.6 WR

struct { ... } WR

WWDR Window register (WWDG_WR)

The documentation for this struct was generated from the following file:

- /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h

Chapter 7

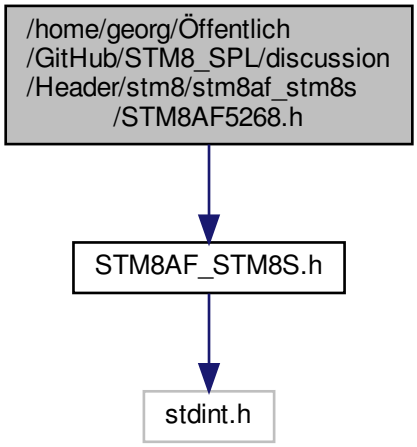
File Documentation

7.1 pages/mainpage.txt File Reference

7.2 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/↩ STM8AF5268.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF5268.h:



Macros

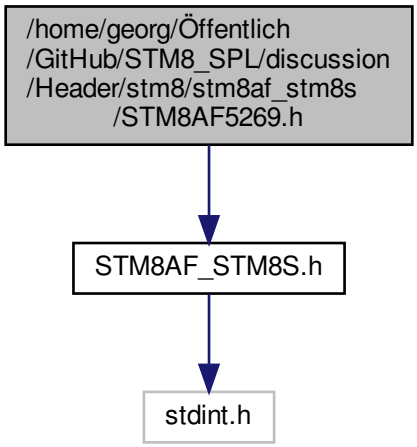
- #define STM8AF5268
- #define STM8AF526x
- #define STM8_PFLASH_SIZE 32768

- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.3 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF5269.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF5269.h:



Macros

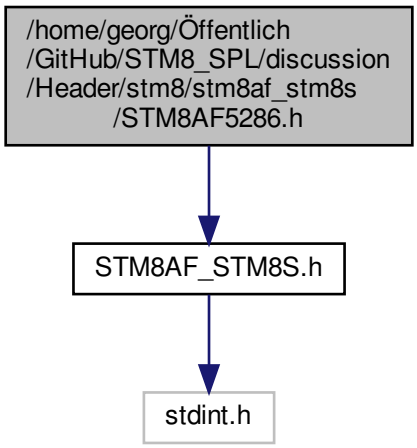
- #define STM8AF5269
- #define STM8AF526x
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.4 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF5286.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF5286.h:



Macros

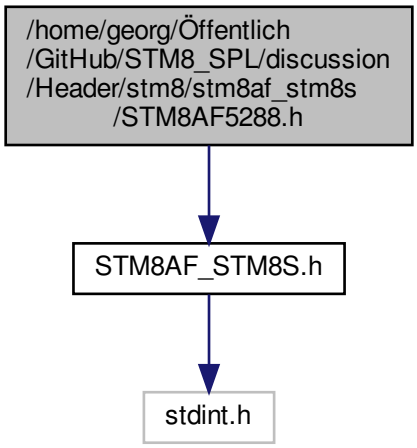
- #define STM8AF5286
- #define STM8AF528x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E

- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.5 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF5288.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF5288.h:



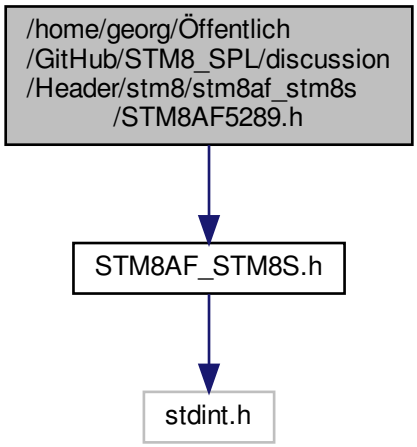
Macros

- #define STM8AF5288
- #define STM8AF528x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.6 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/↩ STM8AF5289.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF5289.h:



Macros

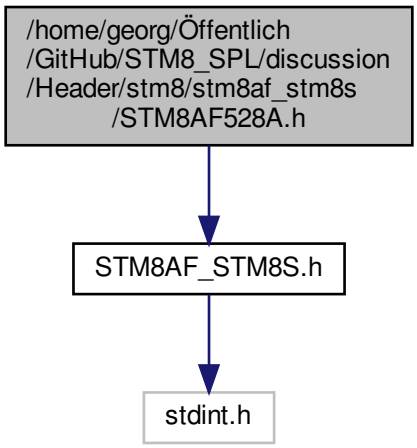
- #define STM8AF5289
- #define STM8AF528x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.7 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF528A.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF528A.h:



Macros

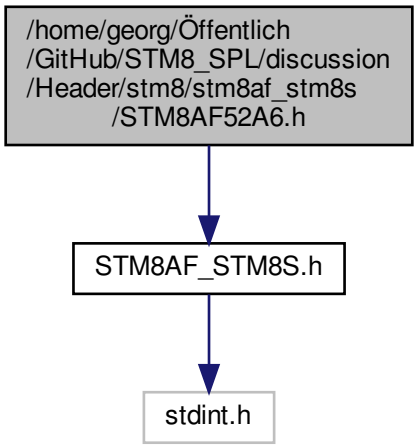
- #define STM8AF528A
- #define STM8AF528x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E

- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.8 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF52A6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF52A6.h:



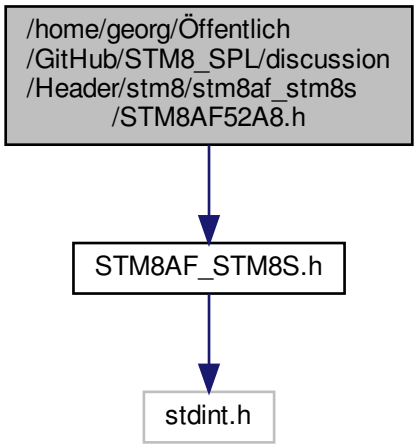
Macros

- #define STM8AF52A6
- #define STM8AF52Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.9 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/↩ STM8AF52A8.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF52A8.h:



Macros

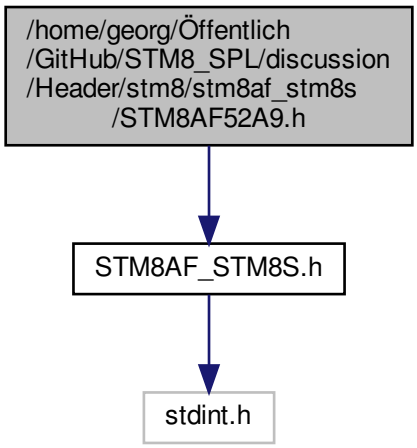
- #define STM8AF52A8
- #define STM8AF52Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.10 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF52A9.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF52A9.h:



Macros

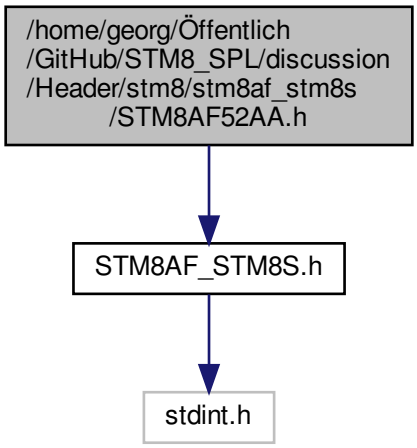
- #define STM8AF52A9
- #define STM8AF52Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E

- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.11 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF52AA.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF52AA.h:



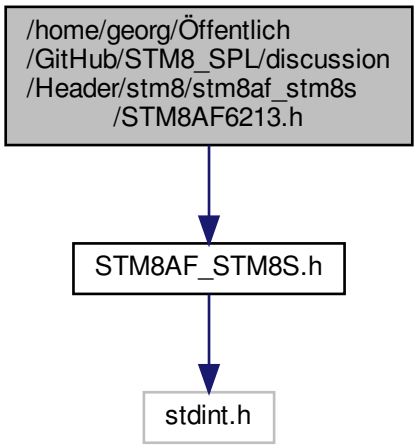
Macros

- #define STM8AF52AA
- #define STM8AF52Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.12 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/↩ STM8AF6213.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6213.h:



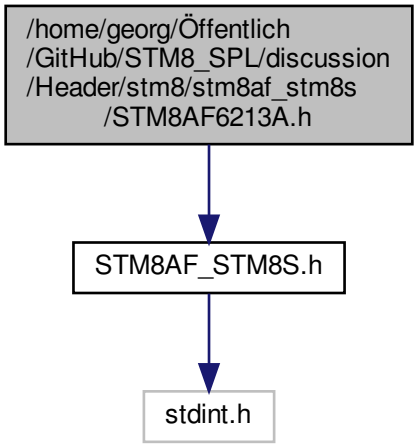
Macros

- #define STM8AF6213
- #define STM8AF621x
- #define STM8_PFLASH_SIZE 4096
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART4_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.13 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6213A.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6213A.h:



Macros

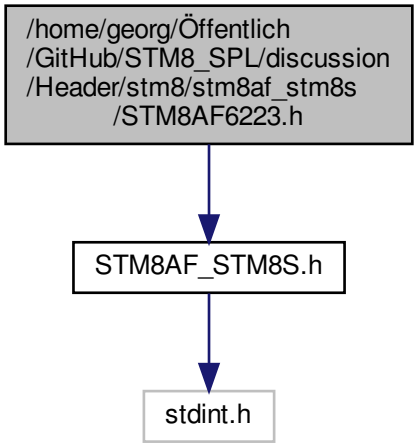
- #define STM8AF6213A
- #define STM8AF621x
- #define STM8_PFLASH_SIZE 4096
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART4_AddressBase 0x5230

- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.14 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6223.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6223.h:



Macros

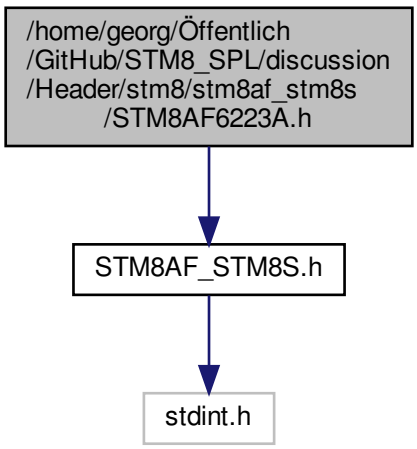
- #define STM8AF6223
- #define STM8AF622x
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0

- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART4_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.15 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6223A.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6223A.h:



Macros

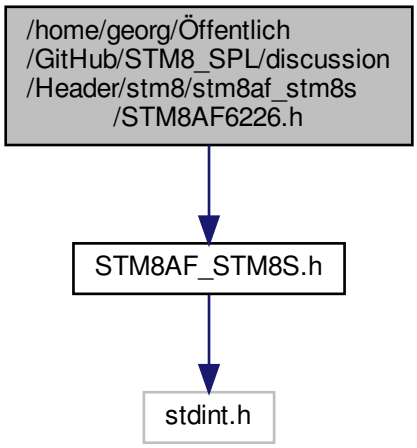
- #define STM8AF6223A
- #define STM8AF622x
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640

- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART4_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.16 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6226.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6226.h:



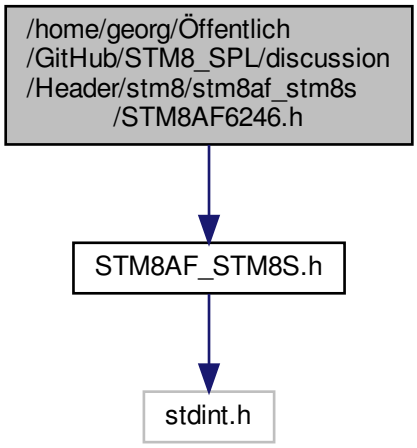
Macros

- #define STM8AF6226
- #define STM8AF622x
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART4_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.17 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/↩ STM8AF6246.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6246.h:



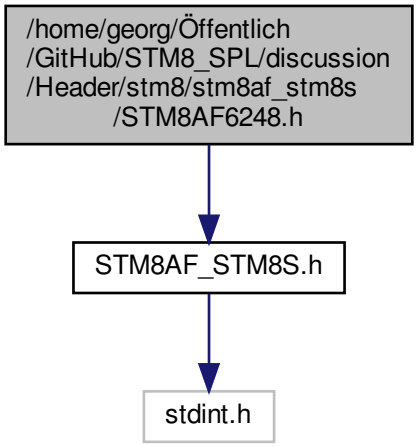
Macros

- #define STM8AF6246
- #define STM8AF624x
- #define STM8_PFLASH_SIZE 16384
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 512
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.18 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6248.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6248.h:



Macros

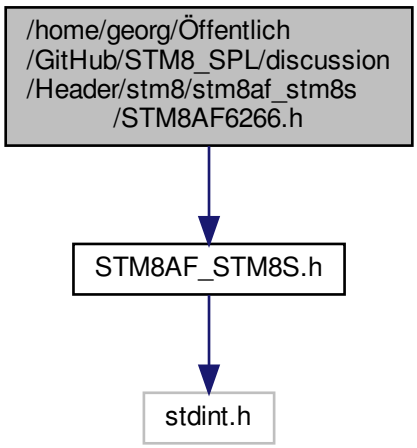
- #define STM8AF6248
- #define STM8AF624x
- #define STM8_PFLASH_SIZE 16384
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 512
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210

- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.19 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6266.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6266.h:



Macros

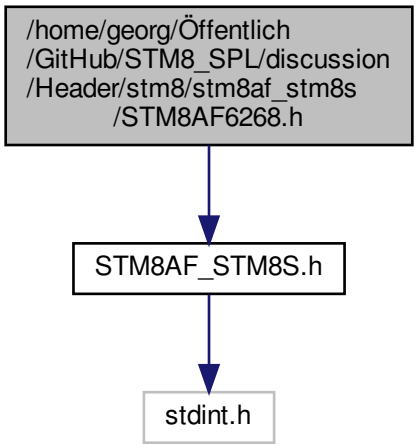
- #define STM8AF6266
- #define STM8AF626x
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019

- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.20 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6268.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6268.h:



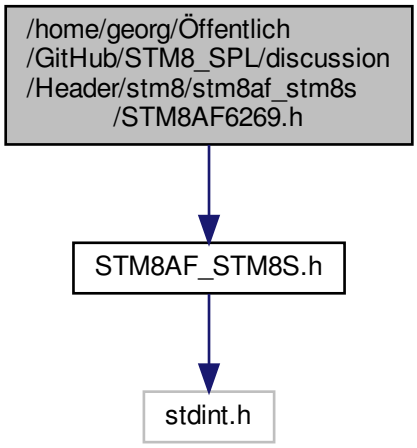
Macros

- #define STM8AF6268
- #define STM8AF626x
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.21 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6269.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6269.h:



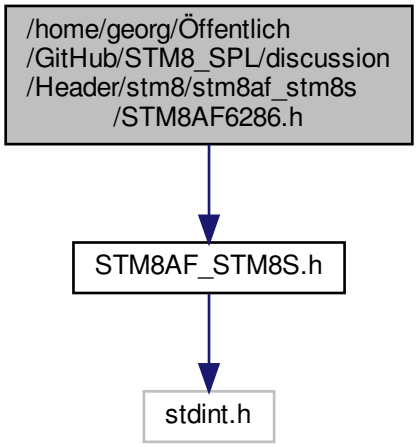
Macros

- #define STM8AF6269
- #define STM8AF626x
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.22 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6286.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6286.h:



Macros

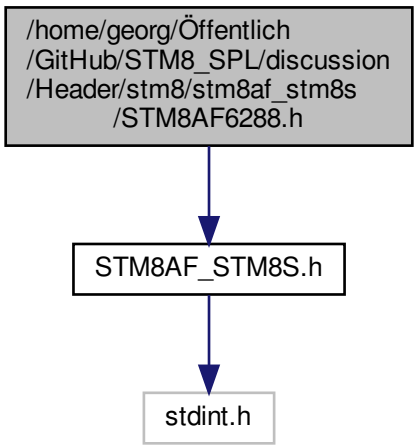
- #define STM8AF6286
- #define STM8AF628x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3

- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.23 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6288.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6288.h:



Macros

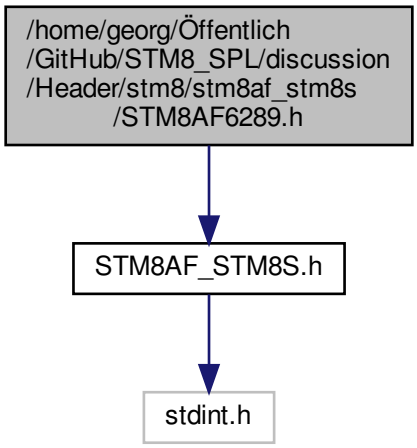
- #define STM8AF6288
- #define STM8AF628x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A

- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.24 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6289.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6289.h:



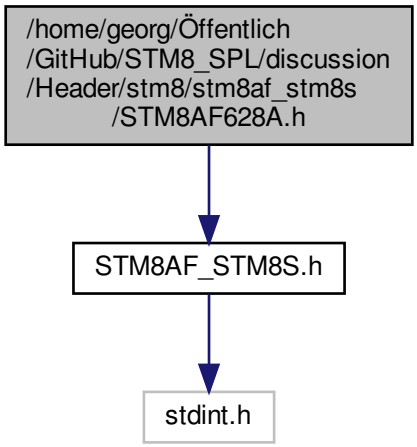
Macros

- #define STM8AF6289
- #define STM8AF628x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.25 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF628A.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF628A.h:



Macros

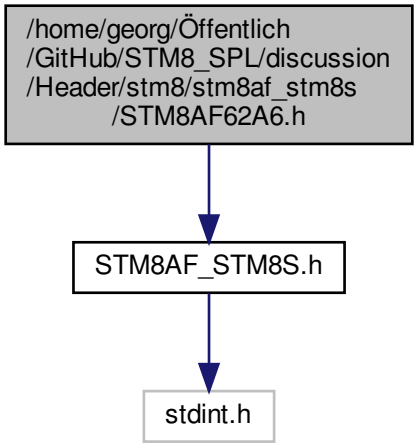
- #define STM8AF628A
- #define STM8AF628x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.26 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF62A6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF62A6.h:



Macros

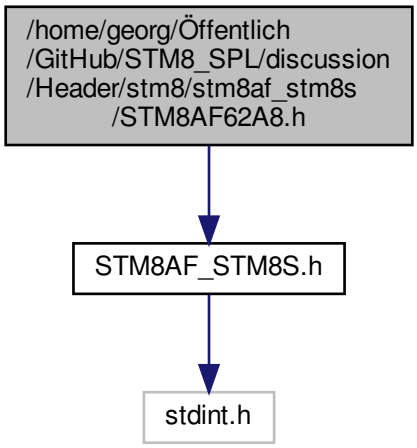
- #define STM8AF62A6
- #define STM8AF62Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023

- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.27 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF62A8.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF62A8.h:



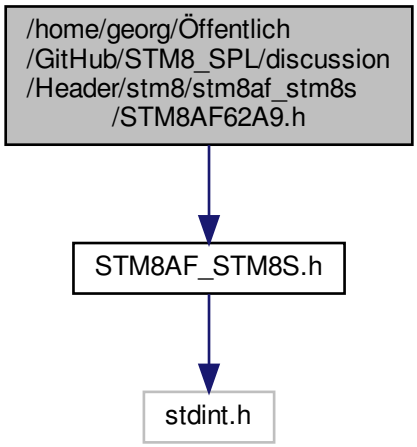
Macros

- #define STM8AF62A8
- #define STM8AF62Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.28 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF62A9.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF62A9.h:



Macros

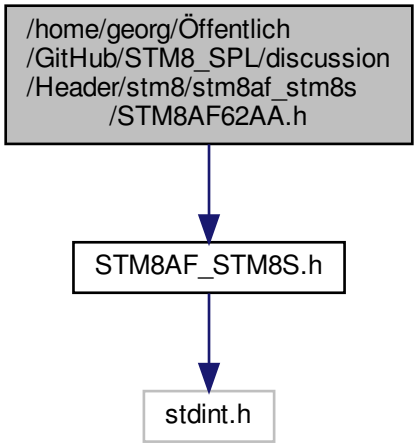
- #define STM8AF62A9
- #define STM8AF62Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.29 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF62AA.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF62AA.h:



Macros

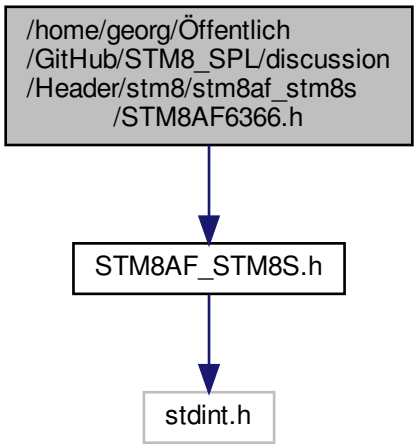
- #define STM8AF62AA
- #define STM8AF62Ax
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023

- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.30 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6366.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6366.h:



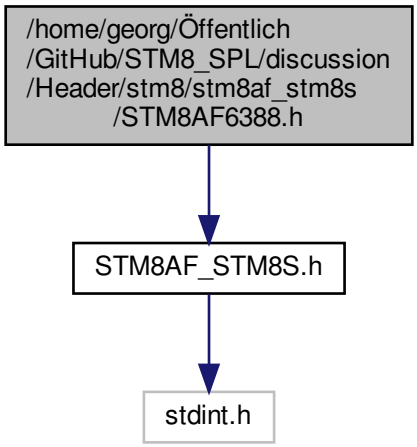
Macros

- #define STM8AF6366
- #define STM8AF636x
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.31 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF6388.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8AF6388.h:



Macros

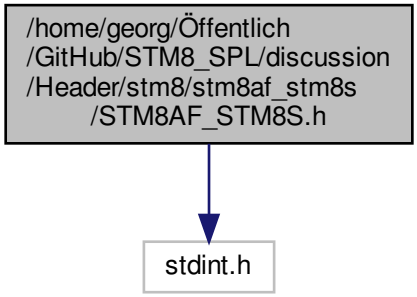
- #define STM8AF6388
- #define STM8AF638x
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.32 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8AF_STM8S.h File Reference

```
#include <stdint.h>
```

Include dependency graph for STM8AF_STM8S.h:



Data Structures

- struct PORT_t
  structure for controlling pins in PORT mode (PORTx, x=A..I)
- struct FLASH_t
  struct to control write/erase of flash memory (FLASH)
- struct EXTI_t
  struct for configuring external port interrupts (EXTI)
- struct RST_t
  struct for determining reset source (RST)
- struct CLK_t
  struct for configuring/monitoring clock module (CLK)
- struct WWDG_t
  struct for access to Window Watchdog registers (WWDG)
- struct IWDG_t
  struct for access to Independent Timeout Watchdog registers (IWDG)
- struct AWU_t
  struct for cofiguring the Auto Wake-Up Module (AWU)
- struct BEEP_t

struct for beeper control (BEEP)

- struct SPI_t

  struct for controlling SPI module (SPI)
- struct I2C_t

  struct for controlling I2C module (I2C)
- struct UART1_t

  struct for controlling Universal Asynchronous Receiver Transmitter 1 (UART1)
- struct UART2_t

  struct for controlling Universal Asynchronous Receiver Transmitter 2 (UART2)
- struct UART3_t

  struct for controlling Universal Asynchronous Receiver Transmitter 3 (UART3)
- struct UART4_t

  struct for controlling Universal Asynchronous Receiver Transmitter 4 (UART4)
- struct TIM1_t

  struct for controlling 16-Bit Timer 1 (TIM1)
- struct TIM2_t

  struct for controlling 16-Bit Timer 2 (TIM2)
- struct TIM3_t

  struct for controlling 16-Bit Timer 3 (TIM3)
- struct TIM4_t

  struct for controlling 8-Bit Timer 4 (TIM4)
- struct TIM5_t

  struct for controlling 16-Bit Timer 5 (TIM5)
- struct TIM6_t

  struct for controlling 8-Bit Timer 6 (TIM6)
- struct ADC1_t

  struct containing Analog Digital Converter 1 (ADC1)
- struct ADC2_t

  struct containing Analog Digital Converter 2 (ADC2)
- struct CAN_t

  struct for controlling Controller Area Network Module (CAN)
- struct CFG_t

  struct for Global Configuration registers (CFG)
- struct ITC_t

  struct for setting interrupt Priority (ITC)

Macros

- #define STM8_PFLASH_SIZE 2048

  size of program flash [B]
- #define STM8_RAM_SIZE 1024

  size of RAM [B]
- #define STM8_EEPROM_SIZE 128

  size of data EEPROM [B]
- #define STM8_PFLASH_START 0x8000

  first address in program flash
- #define STM8_PFLASH_END (STM8_PFLASH_START + STM8_PFLASH_SIZE - 1)

  last address in program flash
- #define STM8_RAM_START 0x0000

  first address in RAM

- #define STM8_RAM_END (STM8_RAM_START + STM8_RAM_SIZE - 1)

  last address in RAM
- #define STM8_EEPROM_START 0x4000

  first address in EEPROM
- #define STM8_EEPROM_END (STM8_EEPROM_START + STM8_EEPROM_SIZE - 1)

  last address in EEPROM
- #define STM8_ADDR_WIDTH 16

  width of address space
- #define STM8_MEM_POINTER_T uint16_t

  address variable type
- #define ISR_HANDLER(func, irq) void func(void) __interrupt(irq)

  handler for interrupt service routine
- #define ISR_HANDLER_TRAP(func) void func() __trap

  handler for trap service routine
- #define NOP() __asm__("nop")

  perform a nop() operation (=minimum delay)
- #define DISABLE_INTERRUPTS() __asm__("sim")

  disable interrupt handling
- #define ENABLE_INTERRUPTS() __asm__("rim")

  enable interrupt handling
- #define TRIGGER_TRAP __asm__("trap")

  trigger a trap (=soft interrupt) e.g. for EMC robustness (see AN1015)
- #define WAIT_FOR_INTERRUPT() __asm__("wfi")

  stop code execution and wait for interrupt
- #define ENTER_HALT() __asm__("halt")

  put controller to HALT mode
- #define SW_RESET() (_WWDG_CR=0xBF)

  reset controller via WWGD module
- #define _BITS unsigned int

  data type in bit structs (follow C90 standard)
- #define _SFR(type, addr) (*((volatile type*) (addr)))

  peripheral register
- #define __TLI_VECTOR__ 0

  irq0 - External Top Level interrupt (TLI) for pin PD7
- #define __AWU_VECTOR__ 1

  irq1 - Auto Wake Up from Halt interrupt (AWU)
- #define __CLK_VECTOR__ 2

  irq2 - Clock Controller interrupt
- #define __PORTA_VECTOR__ 3

  irq3 - External interrupt 0 (GPIOA)
- #define __PORTB_VECTOR__ 4

  irq4 - External interrupt 1 (GPIOB)
- #define __PORTC_VECTOR__ 5

  irq5 - External interrupt 2 (GPIOC)
- #define __PORTD_VECTOR__ 6

  irq6 - External interrupt 3 (GPIOD)
- #define __PORTE_VECTOR__ 7

  irq7 - External interrupt 4 (GPIOE)
- #define __CAN_RX_VECTOR__ 8

  irq8 - CAN receive interrupt (shared with __PORTF_VECTOR__)
- #define __PORTF_VECTOR__ 8

irq8 - External interrupt 5 (GPIOF, shared with __CAN_RX_VECTOR__)

- #define __CAN_TX_VECTOR__ 9

  irq9 - CAN transmit interrupt
- #define __SPI_VECTOR__ 10

  irq10 - SPI End of transfer interrupt
- #define __TIM1_UPD_OVF_VECTOR__ 11

  irq11 - TIM1 Update/Overflow/Trigger/Break interrupt
- #define __TIM1_CAPCOM_VECTOR__ 12

  irq12 - TIM1 Capture/Compare interrupt
- #define __TIM2_UPD_OVF_VECTOR__ 13

  irq13 - TIM2 Update/overflow interrupt (shared with __TIM5_UPD_OVF_VECTOR__)
- #define __TIM5_UPD_OVF_VECTOR__ 13

  irq13 - TIM5 Update/overflow interrupt (shared with __TIM2_UPD_OVF_VECTOR__)
- #define __TIM2_CAPCOM_VECTOR__ 14

  irq14 - TIM2 Capture/Compare interrupt (shared with __TIM5_CAPCOM_VECTOR__)
- #define __TIM3_UPD_OVF_VECTOR__ 15

  irq15 - TIM3 Update/overflow interrupt
- #define __TIM3_CAPCOM_VECTOR__ 16

  irq16 - TIM3 Capture/Compare interrupt
- #define __UART1_TXE_VECTOR__ 17

  irq17 - USART/UART1 send (TX empty) interrupt
- #define __UART1_RXF_VECTOR__ 18

  irq18 - USART/UART1 receive (RX full) interrupt
- #define __I2C_VECTOR__ 19

  irq19 - I2C interrupt
- #define __UART2_TXE_VECTOR__ 20

  irq20 - UART2 send (TX empty) interrupt (shared with __UART3_TXE_VECTOR__ and __UART4_TXE_VECTO↩ R__)
- #define __UART2_RXF_VECTOR__ 21

  irq21 - UART2 receive (RX full) interrupt (shared with __UART3_RXF_VECTOR__ and __UART4_RXF_VECTOR↩ __)
- #define __ADC1_VECTOR__ 22

  irq22 - ADC1 end of conversion (shared with __ADC2_VECTOR__)
- #define __TIM4_UPD_OVF_VECTOR__ 23

  irq23 - TIM4 Update/Overflow interrupt (shared with __TIM6_UPD_OVF_VECTOR__)
- #define __FLASH_VECTOR__ 24

  irq24 - flash interrupt
- #define _GPIOA _SFR(PORT_t, PORTA_AddressBase)

  port A struct/bit access
- #define _GPIOA_ODR _SFR(uint8_t, PORTA_AddressBase+0x00)

  port A output register
- #define _GPIOA_IDR _SFR(uint8_t, PORTA_AddressBase+0x01)

  port A input register
- #define _GPIOA_DDR _SFR(uint8_t, PORTA_AddressBase+0x02)

  port A direction register
- #define _GPIOA_CR1 _SFR(uint8_t, PORTA_AddressBase+0x03)

  port A control register 1
- #define _GPIOA_CR2 _SFR(uint8_t, PORTA_AddressBase+0x04)

  port A control register 2
- #define _GPIOB _SFR(PORT_t, PORTB_AddressBase)

  port B struct/bit access

- #define _GPIOB_ODR _SFR(uint8_t, PORTB_AddressBase+0x00)
  port B output register
- #define _GPIOB_IDR _SFR(uint8_t, PORTB_AddressBase+0x01)
  port B input register
- #define _GPIOB_DDR _SFR(uint8_t, PORTB_AddressBase+0x02)
  port B direction register
- #define _GPIOB_CR1 _SFR(uint8_t, PORTB_AddressBase+0x03)
  port B control register 1
- #define _GPIOB_CR2 _SFR(uint8_t, PORTB_AddressBase+0x04)
  port B control register 2
- #define _GPIOC _SFR(PORT_t, PORTC_AddressBase)
  port C struct/bit access
- #define _GPIOC_ODR _SFR(uint8_t, PORTC_AddressBase+0x00)
  port C output register
- #define _GPIOC_IDR _SFR(uint8_t, PORTC_AddressBase+0x01)
  port C input register
- #define _GPIOC_DDR _SFR(uint8_t, PORTC_AddressBase+0x02)
  port C direction register
- #define _GPIOC_CR1 _SFR(uint8_t, PORTC_AddressBase+0x03)
  port C control register 1
- #define _GPIOC_CR2 _SFR(uint8_t, PORTC_AddressBase+0x04)
  port C control register 2
- #define _GPIOD _SFR(PORT_t, PORTD_AddressBase)
  port D struct/bit access
- #define _GPIOD_ODR _SFR(uint8_t, PORTD_AddressBase+0x00)
  port D output register
- #define _GPIOD_IDR _SFR(uint8_t, PORTD_AddressBase+0x01)
  port D input register
- #define _GPIOD_DDR _SFR(uint8_t, PORTD_AddressBase+0x02)
  port D direction register
- #define _GPIOD_CR1 _SFR(uint8_t, PORTD_AddressBase+0x03)
  port D control register 1
- #define _GPIOD_CR2 _SFR(uint8_t, PORTD_AddressBase+0x04)
  port D control register 2
- #define _GPIOE _SFR(PORT_t, PORTE_AddressBase)
  port E struct/bit access
- #define _GPIOE_ODR _SFR(uint8_t, PORTE_AddressBase+0x00)
  port E output register
- #define _GPIOE_IDR _SFR(uint8_t, PORTE_AddressBase+0x01)
  port E input register
- #define _GPIOE_DDR _SFR(uint8_t, PORTE_AddressBase+0x02)
  port E direction register
- #define _GPIOE_CR1 _SFR(uint8_t, PORTE_AddressBase+0x03)
  port E control register 1
- #define _GPIOE_CR2 _SFR(uint8_t, PORTE_AddressBase+0x04)
  port E control register 2
- #define _GPIOF _SFR(PORT_t, PORTF_AddressBase)
  port F struct/bit access
- #define _GPIOF_ODR _SFR(uint8_t, PORTF_AddressBase+0x00)
  port F output register
- #define _GPIOF_IDR _SFR(uint8_t, PORTF_AddressBase+0x01)

port F input register

- #define _GPIOF_DDR _SFR(uint8_t, PORTF_AddressBase+0x02)

  port F direction register
- #define _GPIOF_CR1 _SFR(uint8_t, PORTF_AddressBase+0x03)

  port F control register 1
- #define _GPIOF_CR2 _SFR(uint8_t, PORTF_AddressBase+0x04)

  port F control register 2
- #define _GPIOG _SFR(PORT_t, PORTG_AddressBase)

  port G struct/bit access
- #define _GPIOG_ODR _SFR(uint8_t, PORTG_AddressBase+0x00)

  port G output register
- #define _GPIOG_IDR _SFR(uint8_t, PORTG_AddressBase+0x01)

  port G input register
- #define _GPIOG_DDR _SFR(uint8_t, PORTG_AddressBase+0x02)

  port G direction register
- #define _GPIOG_CR1 _SFR(uint8_t, PORTG_AddressBase+0x03)

  port G control register 1
- #define _GPIOG_CR2 _SFR(uint8_t, PORTG_AddressBase+0x04)

  port G control register 2
- #define _GPIOH _SFR(PORT_t, PORTH_AddressBase)

  port H struct/bit access
- #define _GPIOH_ODR _SFR(uint8_t, PORTH_AddressBase+0x00)

  port H output register
- #define _GPIOH_IDR _SFR(uint8_t, PORTH_AddressBase+0x01)

  port H input register
- #define _GPIOH_DDR _SFR(uint8_t, PORTH_AddressBase+0x02)

  port H direction register
- #define _GPIOH_CR1 _SFR(uint8_t, PORTH_AddressBase+0x03)

  port H control register 1
- #define _GPIOH_CR2 _SFR(uint8_t, PORTH_AddressBase+0x04)

  port H control register 2
- #define _GPIOI _SFR(PORT_t, PORTI_AddressBase)

  port I struct/bit access
- #define _GPIOI_ODR _SFR(uint8_t, PORTI_AddressBase+0x00)

  port I output register
- #define _GPIOI_IDR _SFR(uint8_t, PORTI_AddressBase+0x01)

  port I input register
- #define _GPIOI_DDR _SFR(uint8_t, PORTI_AddressBase+0x02)

  port I direction register
- #define _GPIOI_CR1 _SFR(uint8_t, PORTI_AddressBase+0x03)

  port I control register 1
- #define _GPIOI_CR2 _SFR(uint8_t, PORTI_AddressBase+0x04)

  port I control register 2
- #define _GPIO_ODR_RESET_VALUE ((uint8_t) 0x00)

  port output register reset value
- #define _GPIO_DDR_RESET_VALUE ((uint8_t) 0x00)

  port direction register reset value
- #define _GPIO_CR1_RESET_VALUE ((uint8_t) 0x00)

  port control register 1 reset value
- #define _GPIO_CR2_RESET_VALUE ((uint8_t) 0x00)

  port control register 2 reset value

- #define _GPIO_PIN0 ((uint8_t) (0x01 << 0))
  port bit mask for pin 0 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN1 ((uint8_t) (0x01 << 1))
  port bit mask for pin 1 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN2 ((uint8_t) (0x01 << 2))
  port bit mask for pin 2 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN3 ((uint8_t) (0x01 << 3))
  port bit mask for pin 3 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN4 ((uint8_t) (0x01 << 4))
  port bit mask for pin 4 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN5 ((uint8_t) (0x01 << 5))
  port bit mask for pin 5 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN6 ((uint8_t) (0x01 << 6))
  port bit mask for pin 6 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN7 ((uint8_t) (0x01 << 7))
  port bit mask for pin 7 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _FLASH _SFR(FLASH_t, FLASH_AddressBase)
  Flash struct/bit access.
- #define _FLASH_CR1 _SFR(uint8_t, FLASH_AddressBase+0x00)
  Flash control register 1 (FLASH_CR1)
- #define _FLASH_CR2 _SFR(uint8_t, FLASH_AddressBase+0x01)
  Flash control register 2 (FLASH_CR2)
- #define _FLASH_NCR2 _SFR(uint8_t, FLASH_AddressBase+0x02)
  complementary Flash control register 2 (FLASH_NCR2)
- #define _FLASH_FPR _SFR(uint8_t, FLASH_AddressBase+0x03)
  Flash protection register (FLASH_FPR)
- #define _FLASH_NFPR _SFR(uint8_t, FLASH_AddressBase+0x04)
  complementary Flash protection register (FLASH_NFPR)
- #define _FLASH_IAPSR _SFR(uint8_t, FLASH_AddressBase+0x05)
  Flash status register (FLASH_IAPSR)
- #define _FLASH_PUKR _SFR(uint8_t, FLASH_AddressBase+0x08)
  Flash program memory unprotecting key register (FLASH_PUKR)
- #define _FLASH_DUKR _SFR(uint8_t, FLASH_AddressBase+0x0A)
  Data EEPROM unprotection key register (FLASH_DUKR)
- #define _FLASH_CR1_RESET_VALUE ((uint8_t) 0x00)
  Flash control register 1 reset value.
- #define _FLASH_CR2_RESET_VALUE ((uint8_t) 0x00)
  Flash control register 2 reset value.
- #define _FLASH_NCR2_RESET_VALUE ((uint8_t) 0xFF)
  complementary Flash control register 2 reset value
- #define _FLASH_IAPSR_RESET_VALUE ((uint8_t) 0x40)
  Flash status register reset value.
- #define _FLASH_PUKR_RESET_VALUE ((uint8_t) 0x00)
  Flash program memory unprotecting key reset value.
- #define _FLASH_DUKR_RESET_VALUE ((uint8_t) 0x00)
  Data EEPROM unprotection key reset value.
- #define _FLASH_FIX ((uint8_t) (0x01 << 0))
  Fixed Byte programming time [0] (in _FLASH_CR1)
- #define _FLASH_IE ((uint8_t) (0x01 << 1))
  Flash Interrupt enable [0] (in _FLASH_CR1)
- #define _FLASH_AHALT ((uint8_t) (0x01 << 2))

Power-down in Active-halt mode [0] (in _FLASH_CR1)

- #define _FLASH_HALT ((uint8_t) (0x01 << 3))

  Power-down in Halt mode [0] (in _FLASH_CR1)
- #define _FLASH_PRG ((uint8_t) (0x01 << 0))

  Standard block programming [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_FPRG ((uint8_t) (0x01 << 4))

  Fast block programming [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_ERASE ((uint8_t) (0x01 << 5))

  Block erasing [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_WPRG ((uint8_t) (0x01 << 6))

  Word programming [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_OPT ((uint8_t) (0x01 << 7))

  Write option bytes [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_WPB ((uint8_t) (0x3F << 0))

  User boot code area protection bits [5:0] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WPB0 ((uint8_t) (0x01 << 0))

  User boot code area protection bit [0] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WPB1 ((uint8_t) (0x01 << 1))

  User boot code area protection bit [1] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WPB2 ((uint8_t) (0x01 << 2))

  User boot code area protection bit [2] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WPB3 ((uint8_t) (0x01 << 3))

  User boot code area protection bit [3] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WPB4 ((uint8_t) (0x01 << 4))

  User boot code area protection bit [4] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WPB5 ((uint8_t) (0x01 << 5))

  User boot code area protection bit [5] (in _FLASH_FPR and _FLASH_NFPR)
- #define _FLASH_WR_PG_DIS ((uint8_t) (0x01 << 0))

  Write attempted to protected page flag [0] (in _FLASH_IAPSR)
- #define _FLASH_PUL ((uint8_t) (0x01 << 1))

  Flash Program memory unlocked flag [0] (in _FLASH_IAPSR)
- #define _FLASH_EOP ((uint8_t) (0x01 << 2))

  End of programming (write or erase operation) flag [0] (in _FLASH_IAPSR)
- #define _FLASH_DUL ((uint8_t) (0x01 << 3))

  Data EEPROM area unlocked flag [0] (in _FLASH_IAPSR)
- #define _FLASH_HVOFF ((uint8_t) (0x01 << 5))

  End of high voltage flag [0] (in _FLASH_IAPSR)
- #define _EXTI _SFR(EXTI_t, EXTI_AddressBase)

  External interrupt struct/bit access.
- #define _EXTI_CR1 _SFR(uint8_t, EXTI_AddressBase+0x00)

  External interrupt control register 1 (EXTI_CR1)
- #define _EXTI_CR2 _SFR(uint8_t, EXTI_AddressBase+0x01)

  External interrupt control register 2 (EXTI_CR2)
- #define _EXTI_CR1_RESET_VALUE ((uint8_t) 0x00)

  External interrupt control register 1 reset value.
- #define _EXTI_CR2_RESET_VALUE ((uint8_t) 0x00)

  External interrupt control register 2 reset value.
- #define _EXTI_PAIS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Port A [1:0] (in _EXTI_CR1)
- #define _EXTI_PAIS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Port A [0] (in _EXTI_CR1)

- #define _EXTI_PAIS1 ((uint8_t) (0x01 << 1))
  External interrupt sensitivity for Port A [1] (in _EXTI_CR1)
- #define _EXTI_PBIS ((uint8_t) (0x03 << 2))
  External interrupt sensitivity for Port B [1:0] (in _EXTI_CR1)
- #define _EXTI_PBIS0 ((uint8_t) (0x01 << 2))
  External interrupt sensitivity for Port B [0] (in _EXTI_CR1)
- #define _EXTI_PBIS1 ((uint8_t) (0x01 << 3))
  External interrupt sensitivity for Port B [1] (in _EXTI_CR1)
- #define _EXTI_PCIS ((uint8_t) (0x03 << 4))
  External interrupt sensitivity for Port C [1:0] (in _EXTI_CR1)
- #define _EXTI_PCIS0 ((uint8_t) (0x01 << 4))
  External interrupt sensitivity for Port C [0] (in _EXTI_CR1)
- #define _EXTI_PCIS1 ((uint8_t) (0x01 << 5))
  External interrupt sensitivity for Port C [1] (in _EXTI_CR1)
- #define _EXTI_PDIS ((uint8_t) (0x03 << 6))
  External interrupt sensitivity for Port D [1:0] (in _EXTI_CR1)
- #define _EXTI_PDIS0 ((uint8_t) (0x01 << 6))
  External interrupt sensitivity for Port D [0] (in _EXTI_CR1)
- #define _EXTI_PDIS1 ((uint8_t) (0x01 << 7))
  External interrupt sensitivity for Port D [1] (in _EXTI_CR1)
- #define _EXTI_PEIS ((uint8_t) (0x03 << 0))
  Port E external interrupt sensitivity bits [1:0] (in _EXTI_CR2)
- #define _EXTI_PEIS0 ((uint8_t) (0x01 << 0))
  Port E external interrupt sensitivity bits [0] (in _EXTI_CR2)
- #define _EXTI_PEIS1 ((uint8_t) (0x01 << 1))
  Port E external interrupt sensitivity bits [1] (in _EXTI_CR2)
- #define _EXTI_TLIS ((uint8_t) (0x01 << 2))
  Top level interrupt sensitivity [0] (in _EXTI_CR2)
- #define _RST _SFR(RST_t, RST_AddressBase)
  Reset module struct/bit access.
- #define _RST_SR _SFR(uint8_t, RST_AddressBase+0x00)
  Reset module status register (RST_SR)
- #define _RST_WWDGF ((uint8_t) (0x01 << 0))
  Window Watchdog reset flag [0] (in _RST_SR)
- #define _RST_IWDGF ((uint8_t) (0x01 << 1))
  Independent Watchdog reset flag [0] (in _RST_SR)
- #define _RST_ILLOPF ((uint8_t) (0x01 << 2))
  Illegal opcode reset flag [0] (in _RST_SR)
- #define _RST_SWIMF ((uint8_t) (0x01 << 3))
  SWIM reset flag [0] (in _RST_SR)
- #define _RST_EMCF ((uint8_t) (0x01 << 4))
  EMC reset flag [0] (in _RST_SR)
- #define _CLK _SFR(CLK_t, CLK_AddressBase)
  Clock module struct/bit access.
- #define _CLK_ICKR _SFR(uint8_t, CLK_AddressBase+0x00)
  Internal clock register.
- #define _CLK_ECKR _SFR(uint8_t, CLK_AddressBase+0x01)
  External clock register.
- #define _CLK_CMSR _SFR(uint8_t, CLK_AddressBase+0x03)
  Clock master status register.
- #define _CLK_SWR _SFR(uint8_t, CLK_AddressBase+0x04)

Clock master switch register.

- #define _CLK_SWCR _SFR(uint8_t, CLK_AddressBase+0x05)

  Clock switch control register.
- #define _CLK_CKDIVR _SFR(uint8_t, CLK_AddressBase+0x06)

  Clock divider register.
- #define _CLK_PCKENR1 _SFR(uint8_t, CLK_AddressBase+0x07)

  Peripheral clock gating register 1.
- #define _CLK_CSSR _SFR(uint8_t, CLK_AddressBase+0x08)

  Clock security system register.
- #define _CLK_CCOR _SFR(uint8_t, CLK_AddressBase+0x09)

  Configurable clock output register.
- #define _CLK_PCKENR2 _SFR(uint8_t, CLK_AddressBase+0x0A)

  Peripheral clock gating register 2.
- #define _CLK_HSITRIMR _SFR(uint8_t, CLK_AddressBase+0x0C)

  HSI clock calibration trimming register.
- #define _CLK_SWIMCCR _SFR(uint8_t, CLK_AddressBase+0x0D)

  SWIM clock control register.
- #define _CLK_ICKR_RESET_VALUE ((uint8_t) 0x01)

  Internal clock register reset value.
- #define _CLK_ECKR_RESET_VALUE ((uint8_t) 0x00)

  External clock register reset value.
- #define _CLK_CMSR_RESET_VALUE ((uint8_t) 0xE1)

  Clock master status reset value.
- #define _CLK_SWR_RESET_VALUE ((uint8_t) 0xE1)

  Clock master switch reset value.
- #define _CLK_SWCR_RESET_VALUE ((uint8_t) 0x00)

  Clock switch control reset value.
- #define _CLK_CKDIVR_RESET_VALUE ((uint8_t) 0x18)

  Clock divider register reset value.
- #define _CLK_PCKENR1_RESET_VALUE ((uint8_t) 0xFF)

  Peripheral clock gating register 1 reset value.
- #define _CLK_PCKENR2_RESET_VALUE ((uint8_t) 0xFF)

  Peripheral clock gating register 2 reset value.
- #define _CLK_CSSR_RESET_VALUE ((uint8_t) 0x00)

  Clock security system register reset value.
- #define _CLK_CCOR_RESET_VALUE ((uint8_t) 0x00)

  Configurable clock output register reset value.
- #define _CLK_HSITRIMR_RESET_VALUE ((uint8_t) 0x00)

  HSI clock calibration trimming register reset value.
- #define _CLK_SWIMCCR_RESET_VALUE ((uint8_t) 0x00)

  SWIM clock control register reset value.
- #define _CLK_HSIEN ((uint8_t) (0x01 << 0))

  High speed internal RC oscillator enable [0] (in _CLK_ICKR)
- #define _CLK_HSIRDY ((uint8_t) (0x01 << 1))

  High speed internal oscillator ready [0] (in _CLK_ICKR)
- #define _CLK_FHWU ((uint8_t) (0x01 << 2))

  Fast wakeup from Halt/Active-halt modes [0] (in _CLK_ICKR)
- #define _CLK_LSIEN ((uint8_t) (0x01 << 3))

  Low speed internal RC oscillator enable [0] (in _CLK_ICKR)
- #define _CLK_LSIRDY ((uint8_t) (0x01 << 4))

  Low speed internal oscillator ready [0] (in _CLK_ICKR)

- #define _CLK_REGAH ((uint8_t) (0x01 << 5))
  Regulator power off in Active-halt mode [0] (in _CLK_ICKR)
- #define _CLK_HSEEN ((uint8_t) (0x01 << 0))
  High speed external crystal oscillator enable [0] (in _CLK_ECKR)
- #define _CLK_ECKR_HSERDY ((uint8_t) (0x01 << 1))
  High speed external crystal oscillator ready [0] (in _CLK_ECKR)
- #define _CLK_SWI_HSI ((uint8_t) 0xE1)
  write to CLK_SWR for HSI clock (in _CLK_SWR)
- #define _CLK_SWI_LSI ((uint8_t) 0xD2)
  write to CLK_SWR for LSI clock (in _CLK_SWR)
- #define _CLK_SWI_HSE ((uint8_t) 0xB4)
  write to CLK_SWR for HSE clock (in _CLK_SWR)
- #define _CLK_SWBSY ((uint8_t) (0x01 << 0))
  Switch busy flag [0] (in _CLK_SWCR)
- #define _CLK_SWEN ((uint8_t) (0x01 << 1))
  Switch start/stop enable [0] (in _CLK_SWCR)
- #define _CLK_SWIEN ((uint8_t) (0x01 << 2))
  Clock switch interrupt enable [0] (in _CLK_SWCR)
- #define _CLK_SWIF ((uint8_t) (0x01 << 3))
  Clock switch interrupt flag [0] (in _CLK_SWCR)
- #define _CLK_CPUDIV ((uint8_t) (0x07 << 0))
  CPU clock prescaler [2:0] (in _CLK_CKDIVR)
- #define _CLK_CPUDIV0 ((uint8_t) (0x01 << 0))
  CPU clock prescaler [0] (in _CLK_CKDIVR)
- #define _CLK_CPUDIV1 ((uint8_t) (0x01 << 1))
  CPU clock prescaler [1] (in _CLK_CKDIVR)
- #define _CLK_CPUDIV2 ((uint8_t) (0x01 << 2))
  CPU clock prescaler [2] (in _CLK_CKDIVR)
- #define _CLK_HSIDIV ((uint8_t) (0x03 << 3))
  High speed internal clock prescaler [1:0] (in _CLK_CKDIVR)
- #define _CLK_HSIDIV0 ((uint8_t) (0x01 << 3))
  High speed internal clock prescaler [0] (in _CLK_CKDIVR)
- #define _CLK_HSIDIV1 ((uint8_t) (0x01 << 4))
  High speed internal clock prescaler [1] (in _CLK_CKDIVR)
- #define _CLK_I2C ((uint8_t) (0x01 << 0))
  clock enable I2C [0] (in _CLK_PCKENR1)
- #define _CLK_SPI ((uint8_t) (0x01 << 1))
  clock enable SPI [0] (in _CLK_PCKENR1)
- #define _CLK_UART1 ((uint8_t) (0x01 << 2))
  clock enable UART1 [0] (in _CLK_PCKENR1)
- #define _CLK_UART2 ((uint8_t) (0x01 << 3))
  clock enable UART2 [0] (in _CLK_PCKENR1)
- #define _CLK_TIM4_TIM6 ((uint8_t) (0x01 << 4))
  clock enable TIM4/TIM6 [0] (in _CLK_PCKENR1)
- #define _CLK_TIM2_TIM5 ((uint8_t) (0x01 << 5))
  clock enable TIM2/TIM5 [0] (in _CLK_PCKENR1)
- #define _CLK_TIM3 ((uint8_t) (0x01 << 6))
  clock enable TIM3 [0] (in _CLK_PCKENR1)
- #define _CLK_TIM1 ((uint8_t) (0x01 << 7))
  clock enable TIM1 [0] (in _CLK_PCKENR1)
- #define _CLK_CSSEN ((uint8_t) (0x01 << 0))

Clock security system enable [0] (in _CLK_CSSR)

- #define _CLK_AUX ((uint8_t) (0x01 << 1))

  Auxiliary oscillator connected to master clock [0] (in _CLK_CSSR)
- #define _CLK_CSSDIE ((uint8_t) (0x01 << 2))

  Clock security system detection interrupt enable [0] (in _CLK_CSSR)
- #define _CLK_CSSD ((uint8_t) (0x01 << 3))

  Clock security system detection [0] (in _CLK_CSSR)
- #define _CLK_CCOEN ((uint8_t) (0x01 << 0))

  Configurable clock output enable [0] (in _CLK_CCOR)
- #define _CLK_CCOSEL ((uint8_t) (0x0F << 1))

  Configurable clock output selection [3:0] (in _CLK_CCOR)
- #define _CLK_CCOSEL0 ((uint8_t) (0x01 << 1))

  Configurable clock output selection [0] (in _CLK_CCOR)
- #define _CLK_CCOSEL1 ((uint8_t) (0x01 << 2))

  Configurable clock output selection [1] (in _CLK_CCOR)
- #define _CLK_CCOSEL2 ((uint8_t) (0x01 << 3))

  Configurable clock output selection [2] (in _CLK_CCOR)
- #define _CLK_CCOSEL3 ((uint8_t) (0x01 << 4))

  Configurable clock output selection [3] (in _CLK_CCOR)
- #define _CLK_CCORDY ((uint8_t) (0x01 << 5))

  Configurable clock output ready [0] (in _CLK_CCOR)
- #define _CLK_CCOBSY ((uint8_t) (0x01 << 6))

  Configurable clock output busy [0] (in _CLK_CCOR)
- #define _CLK_AWU ((uint8_t) (0x01 << 2))

  clock enable AWU [0] (in _CLK_PCKENR2)
- #define _CLK_ADC ((uint8_t) (0x01 << 3))

  clock enable ADC [0] (in _CLK_PCKENR2)
- #define _CLK_CAN ((uint8_t) (0x01 << 7))

  clock enable CAN [0] (in _CLK_PCKENR2)
- #define _CLK_HSITRIM ((uint8_t) (0x0F << 0))

  HSI trimming value (some devices only support 3 bits, see DS!) [3:0] (in _CLK_HSITRIMR)
- #define _CLK_HSITRIM0 ((uint8_t) (0x01 << 0))

  HSI trimming value [0] (in _CLK_HSITRIMR)
- #define _CLK_HSITRIM1 ((uint8_t) (0x01 << 1))

  HSI trimming value [1] (in _CLK_HSITRIMR)
- #define _CLK_HSITRIM2 ((uint8_t) (0x01 << 2))

  HSI trimming value [2] (in _CLK_HSITRIMR)
- #define _CLK_HSITRIM3 ((uint8_t) (0x01 << 3))

  HSI trimming value [3] (in _CLK_HSITRIMR)
- #define _CLK_SWIMCLK ((uint8_t) (0x01 << 0))

  SWIM clock divider [0] (in _CLK_SWIMCCR)
- #define _WWDG _SFR(WWDG_t, WWDG_AddressBase)

  Window Watchdog struct/bit access.
- #define _WWDG_CR _SFR(uint8_t, WWDG_AddressBase+0x00)

  Window Watchdog Control register (WWDG_CR)
- #define _WWDG_WR _SFR(uint8_t, WWDG_AddressBase+0x01)

  Window Watchdog Window register (WWDG_WR)
- #define _WWDG_CR_RESET_VALUE ((uint8_t) 0x7F)

  Window Watchdog Control register reset value.
- #define _WWDG_WR_RESET_VALUE ((uint8_t) 0x7F)

  Window Watchdog Window register reset value.

- #define _WWDG_T ((uint8_t) (0x7F << 0))

  Window Watchdog 7-bit counter [6:0] (in _WWDG_CR)
- #define _WWDG_T0 ((uint8_t) (0x01 << 0))

  Window Watchdog 7-bit counter [0] (in _WWDG_CR)
- #define _WWDG_T1 ((uint8_t) (0x01 << 1))

  Window Watchdog 7-bit counter [1] (in _WWDG_CR)
- #define _WWDG_T2 ((uint8_t) (0x01 << 2))

  Window Watchdog 7-bit counter [2] (in _WWDG_CR)
- #define _WWDG_T3 ((uint8_t) (0x01 << 3))

  Window Watchdog 7-bit counter [3] (in _WWDG_CR)
- #define _WWDG_T4 ((uint8_t) (0x01 << 4))

  Window Watchdog 7-bit counter [4] (in _WWDG_CR)
- #define _WWDG_T5 ((uint8_t) (0x01 << 5))

  Window Watchdog 7-bit counter [5] (in _WWDG_CR)
- #define _WWDG_T6 ((uint8_t) (0x01 << 6))

  Window Watchdog 7-bit counter [6] (in _WWDG_CR)
- #define _WWDG_WDGA ((uint8_t) (0x01 << 7))

  Window Watchdog activation bit (n/a if WWDG enabled by option byte) [0] (in _WWDG_CR)
- #define _WWDG_W ((uint8_t) (0x7F << 0))

  Window Watchdog 7-bit window value [6:0] (in _WWDG_WR)
- #define _WWDG_W0 ((uint8_t) (0x01 << 0))

  Window Watchdog 7-bit window value [0] (in _WWDG_WR)
- #define _WWDG_W1 ((uint8_t) (0x01 << 1))

  Window Watchdog 7-bit window value [1] (in _WWDG_WR)
- #define _WWDG_W2 ((uint8_t) (0x01 << 2))

  Window Watchdog 7-bit window value [2] (in _WWDG_WR)
- #define _WWDG_W3 ((uint8_t) (0x01 << 3))

  Window Watchdog 7-bit window value [3] (in _WWDG_WR)
- #define _WWDG_W4 ((uint8_t) (0x01 << 4))

  Window Watchdog 7-bit window value [4] (in _WWDG_WR)
- #define _WWDG_W5 ((uint8_t) (0x01 << 5))

  Window Watchdog 7-bit window value [5] (in _WWDG_WR)
- #define _WWDG_W6 ((uint8_t) (0x01 << 6))

  Window Watchdog 7-bit window value [6] (in _WWDG_WR)
- #define _IWDG _SFR(IWDG_t, IWDG_AddressBase)

  Independent Timeout Watchdog struct/bit access.
- #define _IWDG_KR _SFR(uint8_t, IWDG_AddressBase+0x00)

  Independent Timeout Watchdog Key register (IWDG_KR)
- #define _IWDG_PR _SFR(uint8_t, IWDG_AddressBase+0x01)

  Independent Timeout Watchdog Prescaler register (IWDG_PR)
- #define _IWDG_RLR _SFR(uint8_t, IWDG_AddressBase+0x02)

  Independent Timeout Watchdog Reload register (IWDG_RLR)
- #define _IWDG_PR_RESET_VALUE ((uint8_t) 0x00)

  Independent Timeout Watchdog Prescaler register reset value.
- #define _IWDG_RLR_RESET_VALUE ((uint8_t) 0xFF)

  Independent Timeout Watchdog Reload register reset value.
- #define _IWDG_KEY_ENABLE ((uint8_t) 0xCC)

  Independent Timeout Watchdog enable (in _IWDG_KR)
- #define _IWDG_KEY_REFRESH ((uint8_t) 0xAA)

  Independent Timeout Watchdog refresh (in _IWDG_KR)
- #define _IWDG_KEY_ACCESS ((uint8_t) 0x55)

Independent Timeout Watchdog unlock write to IWDG_PR and IWDG_RLR (in _IWDG_KR)

- #define _IWDG_PRE ((uint8_t) (0x07 << 0))

  Independent Timeout Watchdog Prescaler divider [2:0] (in _IWDG_PR)
- #define _IWDG_PRE0 ((uint8_t) (0x01 << 0))

  Independent Timeout Watchdog Prescaler divider [0] (in _IWDG_PR)
- #define _IWDG_PRE1 ((uint8_t) (0x01 << 1))

  Independent Timeout Watchdog Prescaler divider [1] (in _IWDG_PR)
- #define _IWDG_PRE2 ((uint8_t) (0x01 << 2))

  Independent Timeout Watchdog Prescaler divider [2] (in _IWDG_PR)
- #define _AWU _SFR(AWU_t, AWU_AddressBase)

  Auto Wake-Up struct/bit access.
- #define _AWU_CSR _SFR(uint8_t, AWU_AddressBase+0x00)

  Auto Wake-Up Control/status register (AWU_CSR)
- #define _AWU_APR _SFR(uint8_t, AWU_AddressBase+0x01)

  Auto Wake-Up Asynchronous prescaler register (AWU_APR)
- #define _AWU_TBR _SFR(uint8_t, AWU_AddressBase+0x02)

  Auto Wake-Up Timebase selection register (AWU_TBR)
- #define _AWU_CSR_RESET_VALUE ((uint8_t) 0x00)

  Auto Wake-Up Control/status register reset value.
- #define _AWU_APR_RESET_VALUE ((uint8_t) 0x3F)

  Auto Wake-Up Asynchronous prescaler register reset value.
- #define _AWU_TBR_RESET_VALUE ((uint8_t) 0x00)

  Auto Wake-Up Timebase selection register reset value.
- #define _AWU_MSR ((uint8_t) (0x01 << 0))

  Auto Wake-Up LSI measurement enable [0] (in _AWU_CSR)
- #define _AWU_AWUEN ((uint8_t) (0x01 << 4))

  Auto-wakeup enable [0] (in _AWU_CSR)
- #define _AWU_AWUF ((uint8_t) (0x01 << 5))

  Auto-wakeup status flag [0] (in _AWU_CSR)
- #define _AWU_APRE ((uint8_t) (0x3F << 0))

  Auto-wakeup asynchronous prescaler divider [5:0] (in _AWU_APR)
- #define _AWU_APRE0 ((uint8_t) (0x01 << 0))

  Auto-wakeup asynchronous prescaler divider [0] (in _AWU_APR)
- #define _AWU_APRE1 ((uint8_t) (0x01 << 1))

  Auto-wakeup asynchronous prescaler divider [1] (in _AWU_APR)
- #define _AWU_APRE2 ((uint8_t) (0x01 << 2))

  Auto-wakeup asynchronous prescaler divider [2] (in _AWU_APR)
- #define _AWU_APRE3 ((uint8_t) (0x01 << 3))

  Auto-wakeup asynchronous prescaler divider [3] (in _AWU_APR)
- #define _AWU_APRE4 ((uint8_t) (0x01 << 4))

  Auto-wakeup asynchronous prescaler divider [4] (in _AWU_APR)
- #define _AWU_APRE5 ((uint8_t) (0x01 << 5))

  Auto-wakeup asynchronous prescaler divider [5] (in _AWU_APR)
- #define _AWU_AWUTB ((uint8_t) (0x0F << 0))

  Auto-wakeup timebase selection [3:0] (in _AWU_APR)
- #define _AWU_AWUTB0 ((uint8_t) (0x01 << 0))

  Auto-wakeup timebase selection [0] (in _AWU_APR)
- #define _AWU_AWUTB1 ((uint8_t) (0x01 << 1))

  Auto-wakeup timebase selection [1] (in _AWU_APR)
- #define _AWU_AWUTB2 ((uint8_t) (0x01 << 2))

  Auto-wakeup timebase selection [2] (in _AWU_APR)

- #define _AWU_AWUTB3 ((uint8_t) (0x01 << 3))
  Auto-wakeup timebase selection [3] (in _AWU_APR)
- #define _BEEP _SFR(BEEP_t, BEEP_AddressBase)
  Beeper struct/bit access.
- #define _BEEP_CSR _SFR(uint8_t, BEEP_AddressBase+0x00)
  Beeper control/status register (BEEP_CSR)
- #define _BEEP_CSR_RESET_VALUE ((uint8_t) 0x1F)
  Beeper control/status register reset value.
- #define _BEEP_BEEPDIV ((uint8_t) (0x1F << 0))
  Beeper clock prescaler divider [4:0] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV0 ((uint8_t) (0x01 << 0))
  Beeper clock prescaler divider [0] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV1 ((uint8_t) (0x01 << 1))
  Beeper clock prescaler divider [1] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV2 ((uint8_t) (0x01 << 2))
  Beeper clock prescaler divider [2] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV3 ((uint8_t) (0x01 << 3))
  Beeper clock prescaler divider [3] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV4 ((uint8_t) (0x01 << 4))
  Beeper clock prescaler divider [4] (in _BEEP_CSR)
- #define _BEEP_BEEPEN ((uint8_t) (0x01 << 5))
  Beeper enable [0] (in _BEEP_CSR)
- #define _BEEP_BEEPSEL ((uint8_t) (0x03 << 6))
  Beeper frequency selection [1:0] (in _BEEP_CSR)
- #define _BEEP_BEEPSEL0 ((uint8_t) (0x01 << 6))
  Beeper frequency selection [0] (in _BEEP_CSR)
- #define _BEEP_BEEPSEL1 ((uint8_t) (0x01 << 7))
  Beeper frequency selection [1] (in _BEEP_CSR)
- #define _SPI _SFR(SPI_t, SPI_AddressBase)
  register for SPI control
- #define _SPI_CR1 _SFR(uint8_t, SPI_AddressBase+0x00)
  SPI control register 1.
- #define _SPI_CR2 _SFR(uint8_t, SPI_AddressBase+0x01)
  SPI control register 2.
- #define _SPI_ICR _SFR(uint8_t, SPI_AddressBase+0x02)
  SPI interrupt control register.
- #define _SPI_SR _SFR(uint8_t, SPI_AddressBase+0x03)
  SPI status register.
- #define _SPI_DR _SFR(uint8_t, SPI_AddressBase+0x04)
  SPI data register.
- #define _SPI_CRCPR _SFR(uint8_t, SPI_AddressBase+0x05)
  SPI CRC polynomial register.
- #define _SPI_RXCRCR _SFR(uint8_t, SPI_AddressBase+0x06)
  SPI Rx CRC register.
- #define _SPI_TXCRCR _SFR(uint8_t, SPI_AddressBase+0x07)
  SPI Tx CRC register.
- #define _SPI_CR1_RESET_VALUE ((uint8_t) 0x00)
  SPI Control Register 1 reset value.
- #define _SPI_CR2_RESET_VALUE ((uint8_t) 0x00)
  SPI Control Register 2 reset value.
- #define _SPI_ICR_RESET_VALUE ((uint8_t) 0x00)

SPI Interrupt Control Register reset value.

- #define _SPI_SR_RESET_VALUE ((uint8_t) 0x02)

  SPI Status Register reset value.
- #define _SPI_DR_RESET_VALUE ((uint8_t) 0x00)

  SPI Data Register reset value.
- #define _SPI_CRCPR_RESET_VALUE ((uint8_t) 0x07)

  SPI Polynomial Register reset value.
- #define _SPI_RXCRCR_RESET_VALUE ((uint8_t) 0x00)

  SPI RX CRC Register reset value.
- #define _SPI_TXCRCR_RESET_VALUE ((uint8_t) 0x00)

  SPI TX CRC Register reset value.
- #define _SPI_CPHA ((uint8_t) (0x01 << 0))

  SPI Clock phase [0] (in _SPI_CR1)
- #define _SPI_CPOL ((uint8_t) (0x01 << 1))

  SPI Clock polarity [0] (in _SPI_CR1)
- #define _SPI_MSTR ((uint8_t) (0x01 << 2))

  SPI Master/slave selection [0] (in _SPI_CR1)
- #define _SPI_BR ((uint8_t) (0x07 << 3))

  SPI Baudrate control [2:0] (in _SPI_CR1)
- #define _SPI_BR0 ((uint8_t) (0x01 << 3))

  SPI Baudrate control [0] (in _SPI_CR1)
- #define _SPI_BR1 ((uint8_t) (0x01 << 4))

  SPI Baudrate control [1] (in _SPI_CR1)
- #define _SPI_BR2 ((uint8_t) (0x01 << 5))

  SPI Baudrate control [2] (in _SPI_CR1)
- #define _SPI_SPE ((uint8_t) (0x01 << 6))

  SPI enable [0] (in _SPI_CR1)
- #define _SPI_LSBFIRST ((uint8_t) (0x01 << 7))

  SPI Frame format [0] (in _SPI_CR1)
- #define _SPI_SSI ((uint8_t) (0x01 << 0))

  SPI Internal slave select [0] (in _SPI_CR2)
- #define _SPI_SSM ((uint8_t) (0x01 << 1))

  SPI Software slave management [0] (in _SPI_CR2)
- #define _SPI_RXONLY ((uint8_t) (0x01 << 2))

  SPI Receive only [0] (in _SPI_CR2)
- #define _SPI_CRCNEXT ((uint8_t) (0x01 << 4))

  SPI Transmit CRC next [0] (in _SPI_CR2)
- #define _SPI_CRCEN ((uint8_t) (0x01 << 5))

  SPI Hardware CRC calculation enable [0] (in _SPI_CR2)
- #define _SPI_BDOE ((uint8_t) (0x01 << 6))

  SPI Input/Output enable in bidirectional mode [0] (in _SPI_CR2)
- #define _SPI_BDM ((uint8_t) (0x01 << 7))

  SPI Bidirectional data mode enable [0] (in _SPI_CR2)
- #define _SPI_WKIE ((uint8_t) (0x01 << 4))

  SPI Wakeup interrupt enable [0] (in _SPI_ICR)
- #define _SPI_ERRIE ((uint8_t) (0x01 << 5))

  SPI Error interrupt enable [0] (in _SPI_ICR)
- #define _SPI_RXIE ((uint8_t) (0x01 << 6))

  SPI Rx buffer not empty interrupt enable [0] (in _SPI_ICR)
- #define _SPI_TXIE ((uint8_t) (0x01 << 7))

  SPI Tx buffer empty interrupt enable [0] (in _SPI_ICR)

- #define _SPI_RXNE ((uint8_t) (0x01 << 0))
  SPI Receive buffer not empty [0] (in _SPI_SR)
- #define _SPI_TXE ((uint8_t) (0x01 << 1))
  SPI Transmit buffer empty [0] (in _SPI_SR)
- #define _SPI_WKUP ((uint8_t) (0x01 << 3))
  SPI Wakeup flag [0] (in _SPI_SR)
- #define _SPI_CRCERR ((uint8_t) (0x01 << 4))
  SPI CRC error flag [0] (in _SPI_SR)
- #define _SPI_MODF ((uint8_t) (0x01 << 5))
  SPI Mode fault [0] (in _SPI_SR)
- #define _SPI_OVR ((uint8_t) (0x01 << 6))
  SPI Overrun flag [0] (in _SPI_SR)
- #define _SPI_BSY ((uint8_t) (0x01 << 7))
  SPI Busy flag [0] (in _SPI_SR)
- #define _I2C _SFR(I2C_t, I2C_AddressBase)
  register for SPI control
- #define _I2C_CR1 _SFR(uint8_t, I2C_AddressBase+0x00)
  I2C Control register 1.
- #define _I2C_CR2 _SFR(uint8_t, I2C_AddressBase+0x01)
  I2C Control register 2.
- #define _I2C_FREQR _SFR(uint8_t, I2C_AddressBase+0x02)
  I2C Frequency register.
- #define _I2C_OARL _SFR(uint8_t, I2C_AddressBase+0x03)
  I2C own address register low byte.
- #define _I2C_OARH _SFR(uint8_t, I2C_AddressBase+0x04)
  I2C own address register high byte.
- #define _I2C_DR _SFR(uint8_t, I2C_AddressBase+0x06)
  I2C data register.
- #define _I2C_SR1 _SFR(uint8_t, I2C_AddressBase+0x07)
  I2C Status register 1.
- #define _I2C_SR2 _SFR(uint8_t, I2C_AddressBase+0x08)
  I2C Status register 2.
- #define _I2C_SR3 _SFR(uint8_t, I2C_AddressBase+0x09)
  I2C Status register 3.
- #define _I2C_ITR _SFR(uint8_t, I2C_AddressBase+0x0A)
  I2C Interrupt register.
- #define _I2C_CCRL _SFR(uint8_t, I2C_AddressBase+0x0B)
  I2C Clock control register low byte.
- #define _I2C_CCRH _SFR(uint8_t, I2C_AddressBase+0x0C)
  I2C Clock control register high byte.
- #define _I2C_TRISER _SFR(uint8_t, I2C_AddressBase+0x0D)
  I2C rise time register.
- #define _I2C_CR1_RESET_VALUE ((uint8_t) 0x00)
  I2C Control register 1 reset value.
- #define _I2C_CR2_RESET_VALUE ((uint8_t) 0x00)
  I2C Control register 2 reset value.
- #define _I2C_FREQR_RESET_VALUE ((uint8_t) 0x00)
  I2C Frequency register reset value.
- #define _I2C_OARL_RESET_VALUE ((uint8_t) 0x00)
  I2C own address register low byte reset value.
- #define _I2C_OARH_RESET_VALUE ((uint8_t) 0x00)

I2C own address register high byte reset value.

- #define _I2C_DR_RESET_VALUE ((uint8_t) 0x00)

  I2C data register reset value.
- #define _I2C_SR1_RESET_VALUE ((uint8_t) 0x00)

  I2C Status register 1 reset value.
- #define _I2C_SR2_RESET_VALUE ((uint8_t) 0x00)

  I2C Status register 2 reset value.
- #define _I2C_SR3_RESET_VALUE ((uint8_t) 0x00)

  I2C Status register 3 reset value.
- #define _I2C_ITR_RESET_VALUE ((uint8_t) 0x00)

  I2C Interrupt register reset value.
- #define _I2C_CCRL_RESET_VALUE ((uint8_t) 0x00)

  I2C Clock control register low byte reset value.
- #define _I2C_CCRH_RESET_VALUE ((uint8_t) 0x00)

  I2C Clock control register high byte reset value.
- #define _I2C_TRISER_RESET_VALUE ((uint8_t) 0x02)

  I2C rise time register reset value.
- #define _I2C_PE ((uint8_t) (0x01 << 0))

  I2C Peripheral enable [0] (in _I2C_CR1)
- #define _I2C_ENGC ((uint8_t) (0x01 << 6))

  I2C General call enable [0] (in _I2C_CR1)
- #define _I2C_NOSTRETCH ((uint8_t) (0x01 << 7))

  I2C Clock stretching disable (Slave mode) [0] (in _I2C_CR1)
- #define _I2C_START ((uint8_t) (0x01 << 0))

  I2C Start generation [0] (in _I2C_CR2)
- #define _I2C_STOP ((uint8_t) (0x01 << 1))

  I2C Stop generation [0] (in _I2C_CR2)
- #define _I2C_ACK ((uint8_t) (0x01 << 2))

  I2C Acknowledge enable [0] (in _I2C_CR2)
- #define _I2C_POS ((uint8_t) (0x01 << 3))

  I2C Acknowledge position (for data reception) [0] (in _I2C_CR2)
- #define _I2C_SWRST ((uint8_t) (0x01 << 7))

  I2C Software reset [0] (in _I2C_CR2)
- #define _I2C_FREQ ((uint8_t) (0x3F << 0))

  I2C Peripheral clock frequency [5:0] (in _I2C_FREQR)
- #define _I2C_FREQ0 ((uint8_t) (0x01 << 0))

  I2C Peripheral clock frequency [0] (in _I2C_FREQR)
- #define _I2C_FREQ1 ((uint8_t) (0x01 << 1))

  I2C Peripheral clock frequency [1] (in _I2C_FREQR)
- #define _I2C_FREQ2 ((uint8_t) (0x01 << 2))

  I2C Peripheral clock frequency [2] (in _I2C_FREQR)
- #define _I2C_FREQ3 ((uint8_t) (0x01 << 3))

  I2C Peripheral clock frequency [3] (in _I2C_FREQR)
- #define _I2C_FREQ4 ((uint8_t) (0x01 << 4))

  I2C Peripheral clock frequency [4] (in _I2C_FREQR)
- #define _I2C_FREQ5 ((uint8_t) (0x01 << 5))

  I2C Peripheral clock frequency [5] (in _I2C_FREQR)
- #define _I2C_ADD0 ((uint8_t) (0x01 << 0))

  I2C Interface address [0] (in 10-bit address mode) (in _I2C_OARL)
- #define _I2C_ADD1 ((uint8_t) (0x01 << 1))

  I2C Interface address [1] (in _I2C_OARL)

- #define _I2C_ADD2 ((uint8_t) (0x01 << 2))
  I2C Interface address [2] (in _I2C_OARL)
- #define _I2C_ADD3 ((uint8_t) (0x01 << 3))
  I2C Interface address [3] (in _I2C_OARL)
- #define _I2C_ADD4 ((uint8_t) (0x01 << 4))
  I2C Interface address [4] (in _I2C_OARL)
- #define _I2C_ADD5 ((uint8_t) (0x01 << 5))
  I2C Interface address [5] (in _I2C_OARL)
- #define _I2C_ADD6 ((uint8_t) (0x01 << 6))
  I2C Interface address [6] (in _I2C_OARL)
- #define _I2C_ADD7 ((uint8_t) (0x01 << 7))
  I2C Interface address [7] (in _I2C_OARL)
- #define _I2C_ADD_8_9 ((uint8_t) (0x03 << 1))
  I2C Interface address [9:8] (in 10-bit address mode) (in _I2C_OARH)
- #define _I2C_ADD8 ((uint8_t) (0x01 << 1))
  I2C Interface address [8] (in _I2C_OARH)
- #define _I2C_ADD9 ((uint8_t) (0x01 << 2))
  I2C Interface address [9] (in _I2C_OARH)
- #define _I2C_ADDCONF ((uint8_t) (0x01 << 6))
  I2C Address mode configuration [0] (in _I2C_OARH)
- #define _I2C_ADDMODE ((uint8_t) (0x01 << 7))
  I2C 7-/10-bit addressing mode (Slave mode) [0] (in _I2C_OARH)
- #define _I2C_SB ((uint8_t) (0x01 << 0))
  I2C Start bit (Master mode) [0] (in _I2C_SR1)
- #define _I2C_ADDR ((uint8_t) (0x01 << 1))
  I2C Address sent (Master mode) / matched (Slave mode) [0] (in _I2C_SR1)
- #define _I2C_BTF ((uint8_t) (0x01 << 2))
  I2C Byte transfer finished [0] (in _I2C_SR1)
- #define _I2C_ADD10 ((uint8_t) (0x01 << 3))
  I2C 10-bit header sent (Master mode) [0] (in _I2C_SR1)
- #define _I2C_STOPF ((uint8_t) (0x01 << 4))
  I2C Stop detection (Slave mode) [0] (in _I2C_SR1)
- #define _I2C_RXNE ((uint8_t) (0x01 << 6))
  I2C Data register not empty (receivers) [0] (in _I2C_SR1)
- #define _I2C_TXE ((uint8_t) (0x01 << 7))
  I2C Data register empty (transmitters) [0] (in _I2C_SR1)
- #define _I2C_BERR ((uint8_t) (0x01 << 0))
  I2C Bus error [0] (in _I2C_SR2)
- #define _I2C_ARLO ((uint8_t) (0x01 << 1))
  I2C Arbitration lost (Master mode) [0] (in _I2C_SR2)
- #define _I2C_AF ((uint8_t) (0x01 << 2))
  I2C Acknowledge failure [0] (in _I2C_SR2)
- #define _I2C_OVR ((uint8_t) (0x01 << 3))
  I2C Overrun/underrun [0] (in _I2C_SR2)
- #define _I2C_WUFH ((uint8_t) (0x01 << 5))
  I2C Wakeup from Halt [0] (in _I2C_SR2)
- #define _I2C_MSL ((uint8_t) (0x01 << 0))
  I2C Master/Slave [0] (in _I2C_SR3)
- #define _I2C_BUSY ((uint8_t) (0x01 << 1))
  I2C Bus busy [0] (in _I2C_SR3)
- #define _I2C_TRA ((uint8_t) (0x01 << 2))

I2C Transmitter/Receiver [0] (in _I2C_SR3)

- #define _I2C_GENCALL ((uint8_t) (0x01 << 4))

  I2C General call header (Slavemode) [0] (in _I2C_SR3)
- #define _I2C_ITERREN ((uint8_t) (0x01 << 0))

  I2C Error interrupt enable [0] (in _I2C_ITR)
- #define _I2C_ITEVTEN ((uint8_t) (0x01 << 1))

  I2C Event interrupt enable [0] (in _I2C_ITR)
- #define _I2C_ITBUFEN ((uint8_t) (0x01 << 2))

  I2C Buffer interrupt enable [0] (in _I2C_ITR)
- #define _I2C_CCR ((uint8_t) (0x0F << 0))

  I2C Clock control register (Master mode) [3:0] (in _I2C_CCRH)
- #define _I2C_CCR0 ((uint8_t) (0x01 << 0))

  I2C Clock control register (Master mode) [0] (in _I2C_CCRH)
- #define _I2C_CCR1 ((uint8_t) (0x01 << 1))

  I2C Clock control register (Master mode) [1] (in _I2C_CCRH)
- #define _I2C_CCR2 ((uint8_t) (0x01 << 2))

  I2C Clock control register (Master mode) [2] (in _I2C_CCRH)
- #define _I2C_CCR3 ((uint8_t) (0x01 << 3))

  I2C Clock control register (Master mode) [3] (in _I2C_CCRH)
- #define _I2C_DUTY ((uint8_t) (0x01 << 6))

  I2C Fast mode duty cycle [0] (in _I2C_CCRH)
- #define _I2C_FS ((uint8_t) (0x01 << 7))

  I2C Master mode selection [0] (in _I2C_CCRH)
- #define _I2C_TRISE ((uint8_t) (0x3F << 0))

  I2C Maximum rise time (Master mode) [5:0] (in _I2C_TRISER)
- #define _I2C_TRISE0 ((uint8_t) (0x01 << 0))

  I2C Maximum rise time (Master mode) [0] (in _I2C_TRISER)
- #define _I2C_TRISE1 ((uint8_t) (0x01 << 1))

  I2C Maximum rise time (Master mode) [1] (in _I2C_TRISER)
- #define _I2C_TRISE2 ((uint8_t) (0x01 << 2))

  I2C Maximum rise time (Master mode) [2] (in _I2C_TRISER)
- #define _I2C_TRISE3 ((uint8_t) (0x01 << 3))

  I2C Maximum rise time (Master mode) [3] (in _I2C_TRISER)
- #define _I2C_TRISE4 ((uint8_t) (0x01 << 4))

  I2C Maximum rise time (Master mode) [4] (in _I2C_TRISER)
- #define _I2C_TRISE5 ((uint8_t) (0x01 << 5))

  I2C Maximum rise time (Master mode) [5] (in _I2C_TRISER)
- #define _UART1 _SFR(UART1_t, UART1_AddressBase)

  UART1 struct/bit access.
- #define _UART1_SR _SFR(uint8_t, UART1_AddressBase+0x00)

  UART1 Status register.
- #define _UART1_DR _SFR(uint8_t, UART1_AddressBase+0x01)

  UART1 data register.
- #define _UART1_BRR1 _SFR(uint8_t, UART1_AddressBase+0x02)

  UART1 Baud rate register 1.
- #define _UART1_BRR2 _SFR(uint8_t, UART1_AddressBase+0x03)

  UART1 Baud rate register 2.
- #define _UART1_CR1 _SFR(uint8_t, UART1_AddressBase+0x04)

  UART1 Control register 1.
- #define _UART1_CR2 _SFR(uint8_t, UART1_AddressBase+0x05)

  UART1 Control register 2.

- #define _UART1_CR3 _SFR(uint8_t, UART1_AddressBase+0x06)
  UART1 Control register 3.
- #define _UART1_CR4 _SFR(uint8_t, UART1_AddressBase+0x07)
  UART1 Control register 4.
- #define _UART1_CR5 _SFR(uint8_t, UART1_AddressBase+0x08)
  UART1 Control register 5.
- #define _UART1_GTR _SFR(uint8_t, UART1_AddressBase+0x09)
  UART1 guard time register.
- #define _UART1_PSCR _SFR(uint8_t, UART1_AddressBase+0x0A)
  UART1 prescaler register.
- #define _UART1_SR_RESET_VALUE ((uint8_t) 0xC0)
  UART1 Status register reset value.
- #define _UART1_BRR1_RESET_VALUE ((uint8_t) 0x00)
  UART1 Baud rate register 1 reset value.
- #define _UART1_BRR2_RESET_VALUE ((uint8_t) 0x00)
  UART1 Baud rate register 2 reset value.
- #define _UART1_CR1_RESET_VALUE ((uint8_t) 0x00)
  UART1 Control register 1 reset value.
- #define _UART1_CR2_RESET_VALUE ((uint8_t) 0x00)
  UART1 Control register 2 reset value.
- #define _UART1_CR3_RESET_VALUE ((uint8_t) 0x00)
  UART1 Control register 3 reset value.
- #define _UART1_CR4_RESET_VALUE ((uint8_t) 0x00)
  UART1 Control register 4 reset value.
- #define _UART1_CR5_RESET_VALUE ((uint8_t) 0x00)
  UART1 Control register 5 reset value.
- #define _UART1_GTR_RESET_VALUE ((uint8_t) 0x00)
  UART1 guard time register reset value.
- #define _UART1_PSCR_RESET_VALUE ((uint8_t) 0x00)
  UART1 prescaler register reset value.
- #define _UART1_PE ((uint8_t) (0x01 << 0))
  UART1 Parity error [0] (in _UART1_SR)
- #define _UART1_FE ((uint8_t) (0x01 << 1))
  UART1 Framing error [0] (in _UART1_SR)
- #define _UART1_NF ((uint8_t) (0x01 << 2))
  UART1 Noise flag [0] (in _UART1_SR)
- #define _UART1_OR_LHE ((uint8_t) (0x01 << 3))
  UART1 LIN Header Error (LIN slave mode) / Overrun error [0] (in _UART1_SR)
- #define _UART1_IDLE ((uint8_t) (0x01 << 4))
  UART1 IDLE line detected [0] (in _UART1_SR)
- #define _UART1_RXNE ((uint8_t) (0x01 << 5))
  UART1 Read data register not empty [0] (in _UART1_SR)
- #define _UART1_TC ((uint8_t) (0x01 << 6))
  UART1 Transmission complete [0] (in _UART1_SR)
- #define _UART1_TXE ((uint8_t) (0x01 << 7))
  UART1 Transmit data register empty [0] (in _UART1_SR)
- #define _UART1_PIEN ((uint8_t) (0x01 << 0))
  UART1 Parity interrupt enable [0] (in _UART1_CR1)
- #define _UART1_PS ((uint8_t) (0x01 << 1))
  UART1 Parity selection [0] (in _UART1_CR1)
- #define _UART1_PCEN ((uint8_t) (0x01 << 2))

UART1 Parity control enable [0] (in _UART1_CR1)

- #define _UART1_WAKE ((uint8_t) (0x01 << 3))

  UART1 Wakeup method [0] (in _UART1_CR1)
- #define _UART1_M ((uint8_t) (0x01 << 4))

  UART1 word length [0] (in _UART1_CR1)
- #define _UART1_UARTD ((uint8_t) (0x01 << 5))

  UART1 Disable (for low power consumption) [0] (in _UART1_CR1)
- #define _UART1_T8 ((uint8_t) (0x01 << 6))

  UART1 Transmit Data bit 8 (in 9-bit mode) [0] (in _UART1_CR1)
- #define _UART1_R8 ((uint8_t) (0x01 << 7))

  UART1 Receive Data bit 8 (in 9-bit mode) [0] (in _UART1_CR1)
- #define _UART1_SBK ((uint8_t) (0x01 << 0))

  UART1 Send break [0] (in _UART1_CR2)
- #define _UART1_RWU ((uint8_t) (0x01 << 1))

  UART1 Receiver wakeup [0] (in _UART1_CR2)
- #define _UART1_REN ((uint8_t) (0x01 << 2))

  UART1 Receiver enable [0] (in _UART1_CR2)
- #define _UART1_TEN ((uint8_t) (0x01 << 3))

  UART1 Transmitter enable [0] (in _UART1_CR2)
- #define _UART1_ILIEN ((uint8_t) (0x01 << 4))

  UART1 IDLE Line interrupt enable [0] (in _UART1_CR2)
- #define _UART1_RIEN ((uint8_t) (0x01 << 5))

  UART1 Receiver interrupt enable [0] (in _UART1_CR2)
- #define _UART1_TCIEN ((uint8_t) (0x01 << 6))

  UART1 Transmission complete interrupt enable [0] (in _UART1_CR2)
- #define _UART1_TIEN ((uint8_t) (0x01 << 7))

  UART1 Transmitter interrupt enable [0] (in _UART1_CR2)
- #define _UART1_LBCL ((uint8_t) (0x01 << 0))

  UART1 Last bit clock pulse [0] (in _UART1_CR3)
- #define _UART1_CPHA ((uint8_t) (0x01 << 1))

  UART1 Clock phase [0] (in _UART1_CR3)
- #define _UART1_CPOL ((uint8_t) (0x01 << 2))

  UART1 Clock polarity [0] (in _UART1_CR3)
- #define _UART1_CKEN ((uint8_t) (0x01 << 3))

  UART1 Clock enable [0] (in _UART1_CR3)
- #define _UART1_STOP ((uint8_t) (0x03 << 4))

  UART1 STOP bits [1:0] (in _UART1_CR3)
- #define _UART1_STOP0 ((uint8_t) (0x01 << 4))

  UART1 STOP bits [0] (in _UART1_CR3)
- #define _UART1_STOP1 ((uint8_t) (0x01 << 5))

  UART1 STOP bits [1] (in _UART1_CR3)
- #define _UART1_LINEN ((uint8_t) (0x01 << 6))

  UART1 LIN mode enable [0] (in _UART1_CR3)
- #define _UART1_ADD ((uint8_t) (0x0F << 0))

  UART1 Address of the UART node [3:0] (in _UART1_CR4)
- #define _UART1_ADD0 ((uint8_t) (0x01 << 0))

  UART1 Address of the UART node [0] (in _UART1_CR4)
- #define _UART1_ADD1 ((uint8_t) (0x01 << 1))

  UART1 Address of the UART node [1] (in _UART1_CR4)
- #define _UART1_ADD2 ((uint8_t) (0x01 << 2))

  UART1 Address of the UART node [2] (in _UART1_CR4)

- #define _UART1_ADD3 ((uint8_t) (0x01 << 3))
  UART1 Address of the UART node [3] (in _UART1_CR4)
- #define _UART1_LBDF ((uint8_t) (0x01 << 4))
  UART1 LIN Break Detection Flag [0] (in _UART1_CR4)
- #define _UART1_LBDL ((uint8_t) (0x01 << 5))
  UART1 LIN Break Detection Length [0] (in _UART1_CR4)
- #define _UART1_LBDIEN ((uint8_t) (0x01 << 6))
  UART1 LIN Break Detection Interrupt Enable [0] (in _UART1_CR4)
- #define _UART1_IREN ((uint8_t) (0x01 << 1))
  UART1 IrDA mode Enable [0] (in _UART1_CR5)
- #define _UART1_IRLP ((uint8_t) (0x01 << 2))
  UART1 IrDA Low Power [0] (in _UART1_CR5)
- #define _UART1_HDSEL ((uint8_t) (0x01 << 3))
  UART1 Half-Duplex Selection [0] (in _UART1_CR5)
- #define _UART1_NACK ((uint8_t) (0x01 << 4))
  UART1 Smartcard NACK enable [0] (in _UART1_CR5)
- #define _UART1_SCEN ((uint8_t) (0x01 << 5))
  UART1 Smartcard mode enable [0] (in _UART1_CR5)
- #define _UART2 _SFR(UART2_t, UART2_AddressBase)
  UART2 struct/bit access.
- #define _UART2_SR _SFR(uint8_t, UART2_AddressBase+0x00)
  UART2 Status register.
- #define _UART2_DR _SFR(uint8_t, UART2_AddressBase+0x01)
  UART2 data register.
- #define _UART2_BRR1 _SFR(uint8_t, UART2_AddressBase+0x02)
  UART2 Baud rate register 1.
- #define _UART2_BRR2 _SFR(uint8_t, UART2_AddressBase+0x03)
  UART2 Baud rate register 2.
- #define _UART2_CR1 _SFR(uint8_t, UART2_AddressBase+0x04)
  UART2 Control register 1.
- #define _UART2_CR2 _SFR(uint8_t, UART2_AddressBase+0x05)
  UART2 Control register 2.
- #define _UART2_CR3 _SFR(uint8_t, UART2_AddressBase+0x06)
  UART2 Control register 3.
- #define _UART2_CR4 _SFR(uint8_t, UART2_AddressBase+0x07)
  UART2 Control register 4.
- #define _UART2_CR5 _SFR(uint8_t, UART2_AddressBase+0x08)
  UART2 Control register 5.
- #define _UART2_CR6 _SFR(uint8_t, UART2_AddressBase+0x09)
  UART2 Control register 6.
- #define _UART2_GTR _SFR(uint8_t, UART2_AddressBase+0x0A)
  UART2 guard time register.
- #define _UART2_PSCR _SFR(uint8_t, UART2_AddressBase+0x0B)
  UART2 prescaler register.
- #define _UART2_SR_RESET_VALUE ((uint8_t) 0xC0)
  UART2 Status register reset value.
- #define _UART2_BRR1_RESET_VALUE ((uint8_t) 0x00)
  UART2 Baud rate register 1 reset value.
- #define _UART2_BRR2_RESET_VALUE ((uint8_t) 0x00)
  UART2 Baud rate register 2 reset value.
- #define _UART2_CR1_RESET_VALUE ((uint8_t) 0x00)

UART2 Control register 1 reset value.

- #define _UART2_CR2_RESET_VALUE ((uint8_t) 0x00)

  UART2 Control register 2 reset value.
- #define _UART2_CR3_RESET_VALUE ((uint8_t) 0x00)

  UART2 Control register 3 reset value.
- #define _UART2_CR4_RESET_VALUE ((uint8_t) 0x00)

  UART2 Control register 4 reset value.
- #define _UART2_CR5_RESET_VALUE ((uint8_t) 0x00)

  UART2 Control register 5 reset value.
- #define _UART2_CR6_RESET_VALUE ((uint8_t) 0x00)

  UART2 Control register 6 reset value.
- #define _UART2_GTR_RESET_VALUE ((uint8_t) 0x00)

  UART2 guard time register reset value.
- #define _UART2_PSCR_RESET_VALUE ((uint8_t) 0x00)

  UART2 prescaler register reset value.
- #define _UART2_PE ((uint8_t) (0x01 << 0))

  UART2 Parity error [0] (in _UART2_SR)
- #define _UART2_FE ((uint8_t) (0x01 << 1))

  UART2 Framing error [0] (in _UART2_SR)
- #define _UART2_NF ((uint8_t) (0x01 << 2))

  UART2 Noise flag [0] (in _UART2_SR)
- #define _UART2_OR_LHE ((uint8_t) (0x01 << 3))

  UART2 LIN Header Error (LIN slave mode) / Overrun error [0] (in _UART2_SR)
- #define _UART2_IDLE ((uint8_t) (0x01 << 4))

  UART2 IDLE line detected [0] (in _UART2_SR)
- #define _UART2_RXNE ((uint8_t) (0x01 << 5))

  UART2 Read data register not empty [0] (in _UART2_SR)
- #define _UART2_TC ((uint8_t) (0x01 << 6))

  UART2 Transmission complete [0] (in _UART2_SR)
- #define _UART2_TXE ((uint8_t) (0x01 << 7))

  UART2 Transmit data register empty [0] (in _UART2_SR)
- #define _UART2_PIEN ((uint8_t) (0x01 << 0))

  UART2 Parity interrupt enable [0] (in _UART2_CR1)
- #define _UART2_PS ((uint8_t) (0x01 << 1))

  UART2 Parity selection [0] (in _UART2_CR1)
- #define _UART2_PCEN ((uint8_t) (0x01 << 2))

  UART2 Parity control enable [0] (in _UART2_CR1)
- #define _UART2_WAKE ((uint8_t) (0x01 << 3))

  UART2 Wakeup method [0] (in _UART2_CR1)
- #define _UART2_M ((uint8_t) (0x01 << 4))

  UART2 word length [0] (in _UART2_CR1)
- #define _UART2_UARTD ((uint8_t) (0x01 << 5))

  UART2 Disable (for low power consumption) [0] (in _UART2_CR1)
- #define _UART2_T8 ((uint8_t) (0x01 << 6))

  UART2 Transmit Data bit 8 (in 9-bit mode) [0] (in _UART2_CR1)
- #define _UART2_R8 ((uint8_t) (0x01 << 7))

  UART2 Receive Data bit 8 (in 9-bit mode) [0] (in _UART2_CR1)
- #define _UART2_SBK ((uint8_t) (0x01 << 0))

  UART2 Send break [0] (in _UART2_CR2)
- #define _UART2_RWU ((uint8_t) (0x01 << 1))

  UART2 Receiver wakeup [0] (in _UART2_CR2)

- #define _UART2_REN ((uint8_t) (0x01 << 2))
  UART2 Receiver enable [0] (in _UART2_CR2)
- #define _UART2_TEN ((uint8_t) (0x01 << 3))
  UART2 Transmitter enable [0] (in _UART2_CR2)
- #define _UART2_ILIEN ((uint8_t) (0x01 << 4))
  UART2 IDLE Line interrupt enable [0] (in _UART2_CR2)
- #define _UART2_RIEN ((uint8_t) (0x01 << 5))
  UART2 Receiver interrupt enable [0] (in _UART2_CR2)
- #define _UART2_TCIEN ((uint8_t) (0x01 << 6))
  UART2 Transmission complete interrupt enable [0] (in _UART2_CR2)
- #define _UART2_TIEN ((uint8_t) (0x01 << 7))
  UART2 Transmitter interrupt enable [0] (in _UART2_CR2)
- #define _UART2_LBCL ((uint8_t) (0x01 << 0))
  UART2 Last bit clock pulse [0] (in _UART2_CR3)
- #define _UART2_CPHA ((uint8_t) (0x01 << 1))
  UART2 Clock phase [0] (in _UART2_CR3)
- #define _UART2_CPOL ((uint8_t) (0x01 << 2))
  UART2 Clock polarity [0] (in _UART2_CR3)
- #define _UART2_CKEN ((uint8_t) (0x01 << 3))
  UART2 Clock enable [0] (in _UART2_CR3)
- #define _UART2_STOP ((uint8_t) (0x03 << 4))
  UART2 STOP bits [1:0] (in _UART2_CR3)
- #define _UART2_STOP0 ((uint8_t) (0x01 << 4))
  UART2 STOP bits [0] (in _UART2_CR3)
- #define _UART2_STOP1 ((uint8_t) (0x01 << 5))
  UART2 STOP bits [1] (in _UART2_CR3)
- #define _UART2_LINEN ((uint8_t) (0x01 << 6))
  UART2 LIN mode enable [0] (in _UART2_CR3)
- #define _UART2_ADD ((uint8_t) (0x0F << 0))
  UART2 Address of the UART node [3:0] (in _UART2_CR4)
- #define _UART2_ADD0 ((uint8_t) (0x01 << 0))
  UART2 Address of the UART node [0] (in _UART2_CR4)
- #define _UART2_ADD1 ((uint8_t) (0x01 << 1))
  UART2 Address of the UART node [1] (in _UART2_CR4)
- #define _UART2_ADD2 ((uint8_t) (0x01 << 2))
  UART2 Address of the UART node [2] (in _UART2_CR4)
- #define _UART2_ADD3 ((uint8_t) (0x01 << 3))
  UART2 Address of the UART node [3] (in _UART2_CR4)
- #define _UART2_LBDF ((uint8_t) (0x01 << 4))
  UART2 LIN Break Detection Flag [0] (in _UART2_CR4)
- #define _UART2_LBDL ((uint8_t) (0x01 << 5))
  UART2 LIN Break Detection Length [0] (in _UART2_CR4)
- #define _UART2_LBDIEN ((uint8_t) (0x01 << 6))
  UART2 LIN Break Detection Interrupt Enable [0] (in _UART2_CR4)
- #define _UART2_IREN ((uint8_t) (0x01 << 1))
  UART2 IrDA mode Enable [0] (in _UART2_CR5)
- #define _UART2_IRLP ((uint8_t) (0x01 << 2))
  UART2 IrDA Low Power [0] (in _UART2_CR5)
- #define _UART2_NACK ((uint8_t) (0x01 << 4))
  UART2 Smartcard NACK enable [0] (in _UART2_CR5)
- #define _UART2_SCEN ((uint8_t) (0x01 << 5))

UART2 Smartcard mode enable [0] (in _UART2_CR5)

- #define _UART2_LSF ((uint8_t) (0x01 << 0))

  UART2 LIN Sync Field [0] (in _UART2_CR6)
- #define _UART2_LHDF ((uint8_t) (0x01 << 1))

  UART2 LIN Header Detection Flag [0] (in _UART2_CR6)
- #define _UART2_LHDIEN ((uint8_t) (0x01 << 2))

  UART2 LIN Header Detection Interrupt Enable [0] (in _UART2_CR6)
- #define _UART2_LASE ((uint8_t) (0x01 << 4))

  UART2 LIN automatic resynchronisation enable [0] (in _UART2_CR6)
- #define _UART2_LSLV ((uint8_t) (0x01 << 5))

  UART2 LIN Slave Enable [0] (in _UART2_CR6)
- #define _UART2_LDUM ((uint8_t) (0x01 << 7))

  UART2 LIN Divider Update Method [0] (in _UART2_CR6)
- #define _UART3 _SFR(UART3_t, UART3_AddressBase)

  UART3 struct/bit access.
- #define _UART3_SR _SFR(uint8_t, UART3_AddressBase+0x00)

  UART3 Status register.
- #define _UART3_DR _SFR(uint8_t, UART3_AddressBase+0x01)

  UART3 data register.
- #define _UART3_BRR1 _SFR(uint8_t, UART3_AddressBase+0x02)

  UART3 Baud rate register 1.
- #define _UART3_BRR2 _SFR(uint8_t, UART3_AddressBase+0x03)

  UART3 Baud rate register 2.
- #define _UART3_CR1 _SFR(uint8_t, UART3_AddressBase+0x04)

  UART3 Control register 1.
- #define _UART3_CR2 _SFR(uint8_t, UART3_AddressBase+0x05)

  UART3 Control register 2.
- #define _UART3_CR3 _SFR(uint8_t, UART3_AddressBase+0x06)

  UART3 Control register 3.
- #define _UART3_CR4 _SFR(uint8_t, UART3_AddressBase+0x07)

  UART3 Control register 4.
- #define _UART3_CR6 _SFR(uint8_t, UART3_AddressBase+0x09)

  UART3 Control register 6.
- #define _UART3_SR_RESET_VALUE ((uint8_t) 0xC0)

  UART3 Status register reset value.
- #define _UART3_BRR1_RESET_VALUE ((uint8_t) 0x00)

  UART3 Baud rate register 1 reset value.
- #define _UART3_BRR2_RESET_VALUE ((uint8_t) 0x00)

  UART3 Baud rate register 2 reset value.
- #define _UART3_CR1_RESET_VALUE ((uint8_t) 0x00)

  UART3 Control register 1 reset value.
- #define _UART3_CR2_RESET_VALUE ((uint8_t) 0x00)

  UART3 Control register 2 reset value.
- #define _UART3_CR3_RESET_VALUE ((uint8_t) 0x00)

  UART3 Control register 3 reset value.
- #define _UART3_CR4_RESET_VALUE ((uint8_t) 0x00)

  UART3 Control register 4 reset value.
- #define _UART3_CR6_RESET_VALUE ((uint8_t) 0x00)

  UART3 Control register 6 reset value.
- #define _UART3_PE ((uint8_t) (0x01 << 0))

  UART3 Parity error [0] (in _UART3_SR)

- #define _UART3_FE ((uint8_t) (0x01 << 1))
  UART3 Framing error [0] (in _UART3_SR)
- #define _UART3_NF ((uint8_t) (0x01 << 2))
  UART3 Noise flag [0] (in _UART3_SR)
- #define _UART3_OR_LHE ((uint8_t) (0x01 << 3))
  UART3 LIN Header Error (LIN slave mode) / Overrun error [0] (in _UART3_SR)
- #define _UART3_IDLE ((uint8_t) (0x01 << 4))
  UART3 IDLE line detected [0] (in _UART3_SR)
- #define _UART3_RXNE ((uint8_t) (0x01 << 5))
  UART3 Read data register not empty [0] (in _UART3_SR)
- #define _UART3_TC ((uint8_t) (0x01 << 6))
  UART3 Transmission complete [0] (in _UART3_SR)
- #define _UART3_TXE ((uint8_t) (0x01 << 7))
  UART3 Transmit data register empty [0] (in _UART3_SR)
- #define _UART3_PIEN ((uint8_t) (0x01 << 0))
  UART3 Parity interrupt enable [0] (in _UART3_CR1)
- #define _UART3_PS ((uint8_t) (0x01 << 1))
  UART3 Parity selection [0] (in _UART3_CR1)
- #define _UART3_PCEN ((uint8_t) (0x01 << 2))
  UART3 Parity control enable [0] (in _UART3_CR1)
- #define _UART3_WAKE ((uint8_t) (0x01 << 3))
  UART3 Wakeup method [0] (in _UART3_CR1)
- #define _UART3_M ((uint8_t) (0x01 << 4))
  UART3 word length [0] (in _UART3_CR1)
- #define _UART3_UARTD ((uint8_t) (0x01 << 5))
  UART3 Disable (for low power consumption) [0] (in _UART3_CR1)
- #define _UART3_T8 ((uint8_t) (0x01 << 6))
  UART3 Transmit Data bit 8 (in 9-bit mode) [0] (in _UART3_CR1)
- #define _UART3_R8 ((uint8_t) (0x01 << 7))
  UART3 Receive Data bit 8 (in 9-bit mode) [0] (in _UART3_CR1)
- #define _UART3_SBK ((uint8_t) (0x01 << 0))
  UART3 Send break [0] (in _UART3_CR2)
- #define _UART3_RWU ((uint8_t) (0x01 << 1))
  UART3 Receiver wakeup [0] (in _UART3_CR2)
- #define _UART3_REN ((uint8_t) (0x01 << 2))
  UART3 Receiver enable [0] (in _UART3_CR2)
- #define _UART3_TEN ((uint8_t) (0x01 << 3))
  UART3 Transmitter enable [0] (in _UART3_CR2)
- #define _UART3_ILIEN ((uint8_t) (0x01 << 4))
  UART3 IDLE Line interrupt enable [0] (in _UART3_CR2)
- #define _UART3_RIEN ((uint8_t) (0x01 << 5))
  UART3 Receiver interrupt enable [0] (in _UART3_CR2)
- #define _UART3_TCIEN ((uint8_t) (0x01 << 6))
  UART3 Transmission complete interrupt enable [0] (in _UART3_CR2)
- #define _UART3_TIEN ((uint8_t) (0x01 << 7))
  UART3 Transmitter interrupt enable [0] (in _UART3_CR2)
- #define _UART3_STOP ((uint8_t) (0x03 << 4))
  UART3 STOP bits [1:0] (in _UART3_CR3)
- #define _UART3_STOP0 ((uint8_t) (0x01 << 4))
  UART3 STOP bits [0] (in _UART3_CR3)
- #define _UART3_STOP1 ((uint8_t) (0x01 << 5))

UART3 STOP bits [1] (in _UART3_CR3)

- #define _UART3_LINEN ((uint8_t) (0x01 << 6))

  UART3 LIN mode enable [0] (in _UART3_CR3)
- #define _UART3_ADD ((uint8_t) (0x0F << 0))

  UART3 Address of the UART node [3:0] (in _UART3_CR4)
- #define _UART3_ADD0 ((uint8_t) (0x01 << 0))

  UART3 Address of the UART node [0] (in _UART3_CR4)
- #define _UART3_ADD1 ((uint8_t) (0x01 << 1))

  UART3 Address of the UART node [1] (in _UART3_CR4)
- #define _UART3_ADD2 ((uint8_t) (0x01 << 2))

  UART3 Address of the UART node [2] (in _UART3_CR4)
- #define _UART3_ADD3 ((uint8_t) (0x01 << 3))

  UART3 Address of the UART node [3] (in _UART3_CR4)
- #define _UART3_LBDF ((uint8_t) (0x01 << 4))

  UART3 LIN Break Detection Flag [0] (in _UART3_CR4)
- #define _UART3_LBDL ((uint8_t) (0x01 << 5))

  UART3 LIN Break Detection Length [0] (in _UART3_CR4)
- #define _UART3_LBDIEN ((uint8_t) (0x01 << 6))

  UART3 LIN Break Detection Interrupt Enable [0] (in _UART3_CR4)
- #define _UART3_LSF ((uint8_t) (0x01 << 0))

  UART3 LIN Sync Field [0] (in _UART3_CR6)
- #define _UART3_LHDF ((uint8_t) (0x01 << 1))

  UART3 LIN Header Detection Flag [0] (in _UART3_CR6)
- #define _UART3_LHDIEN ((uint8_t) (0x01 << 2))

  UART3 LIN Header Detection Interrupt Enable [0] (in _UART3_CR6)
- #define _UART3_LASE ((uint8_t) (0x01 << 4))

  UART3 LIN automatic resynchronisation enable [0] (in _UART3_CR6)
- #define _UART3_LSLV ((uint8_t) (0x01 << 5))

  UART3 LIN Slave Enable [0] (in _UART3_CR6)
- #define _UART3_LDUM ((uint8_t) (0x01 << 7))

  UART3 LIN Divider Update Method [0] (in _UART3_CR6)
- #define _UART4 _SFR(UART4_t, UART4_AddressBase)

  UART4 struct/bit access.
- #define _UART4_SR _SFR(uint8_t, UART4_AddressBase+0x00)

  UART4 Status register.
- #define _UART4_DR _SFR(uint8_t, UART4_AddressBase+0x01)

  UART4 data register.
- #define _UART4_BRR1 _SFR(uint8_t, UART4_AddressBase+0x02)

  UART4 Baud rate register 1.
- #define _UART4_BRR2 _SFR(uint8_t, UART4_AddressBase+0x03)

  UART4 Baud rate register 2.
- #define _UART4_CR1 _SFR(uint8_t, UART4_AddressBase+0x04)

  UART4 Control register 1.
- #define _UART4_CR2 _SFR(uint8_t, UART4_AddressBase+0x05)

  UART4 Control register 2.
- #define _UART4_CR3 _SFR(uint8_t, UART4_AddressBase+0x06)

  UART4 Control register 3.
- #define _UART4_CR4 _SFR(uint8_t, UART4_AddressBase+0x07)

  UART4 Control register 4.
- #define _UART4_CR5 _SFR(uint8_t, UART4_AddressBase+0x08)

  UART4 Control register 5.

- #define _UART4_CR6 _SFR(uint8_t, UART4_AddressBase+0x09)
  UART4 Control register 6.
- #define _UART4_GTR _SFR(uint8_t, UART4_AddressBase+0x0A)
  UART4 guard time register.
- #define _UART4_PSCR _SFR(uint8_t, UART4_AddressBase+0x0B)
  UART4 prescaler register.
- #define _UART4_SR_RESET_VALUE ((uint8_t) 0xC0)
  UART4 Status register reset value.
- #define _UART4_BRR1_RESET_VALUE ((uint8_t) 0x00)
  UART4 Baud rate register 1 reset value.
- #define _UART4_BRR2_RESET_VALUE ((uint8_t) 0x00)
  UART4 Baud rate register 2 reset value.
- #define _UART4_CR1_RESET_VALUE ((uint8_t) 0x00)
  UART4 Control register 1 reset value.
- #define _UART4_CR2_RESET_VALUE ((uint8_t) 0x00)
  UART4 Control register 2 reset value.
- #define _UART4_CR3_RESET_VALUE ((uint8_t) 0x00)
  UART4 Control register 3 reset value.
- #define _UART4_CR4_RESET_VALUE ((uint8_t) 0x00)
  UART4 Control register 4 reset value.
- #define _UART4_CR5_RESET_VALUE ((uint8_t) 0x00)
  UART4 Control register 5 reset value.
- #define _UART4_CR6_RESET_VALUE ((uint8_t) 0x00)
  UART4 Control register 6 reset value.
- #define _UART4_GTR_RESET_VALUE ((uint8_t) 0x00)
  UART4 guard time register reset value.
- #define _UART4_PSCR_RESET_VALUE ((uint8_t) 0x00)
  UART4 prescaler register reset value.
- #define _UART4_PE ((uint8_t) (0x01 << 0))
  UART4 Parity error [0] (in _UART4_SR)
- #define _UART4_FE ((uint8_t) (0x01 << 1))
  UART4 Framing error [0] (in _UART4_SR)
- #define _UART4_NF ((uint8_t) (0x01 << 2))
  UART4 Noise flag [0] (in _UART4_SR)
- #define _UART4_OR_LHE ((uint8_t) (0x01 << 3))
  UART4 LIN Header Error (LIN slave mode) / Overrun error [0] (in _UART4_SR)
- #define _UART4_IDLE ((uint8_t) (0x01 << 4))
  UART4 IDLE line detected [0] (in _UART4_SR)
- #define _UART4_RXNE ((uint8_t) (0x01 << 5))
  UART4 Read data register not empty [0] (in _UART4_SR)
- #define _UART4_TC ((uint8_t) (0x01 << 6))
  UART4 Transmission complete [0] (in _UART4_SR)
- #define _UART4_TXE ((uint8_t) (0x01 << 7))
  UART4 Transmit data register empty [0] (in _UART4_SR)
- #define _UART4_PIEN ((uint8_t) (0x01 << 0))
  UART4 Parity interrupt enable [0] (in _UART4_CR1)
- #define _UART4_PS ((uint8_t) (0x01 << 1))
  UART4 Parity selection [0] (in _UART4_CR1)
- #define _UART4_PCEN ((uint8_t) (0x01 << 2))
  UART4 Parity control enable [0] (in _UART4_CR1)
- #define _UART4_WAKE ((uint8_t) (0x01 << 3))

UART4 Wakeup method [0] (in _UART4_CR1)

- #define _UART4_M ((uint8_t) (0x01 << 4))

  UART4 word length [0] (in _UART4_CR1)
- #define _UART4_UARTD ((uint8_t) (0x01 << 5))

  UART4 Disable (for low power consumption) [0] (in _UART4_CR1)
- #define _UART4_T8 ((uint8_t) (0x01 << 6))

  UART4 Transmit Data bit 8 (in 9-bit mode) [0] (in _UART4_CR1)
- #define _UART4_R8 ((uint8_t) (0x01 << 7))

  UART4 Receive Data bit 8 (in 9-bit mode) [0] (in _UART4_CR1)
- #define _UART4_SBK ((uint8_t) (0x01 << 0))

  UART4 Send break [0] (in _UART4_CR2)
- #define _UART4_RWU ((uint8_t) (0x01 << 1))

  UART4 Receiver wakeup [0] (in _UART4_CR2)
- #define _UART4_REN ((uint8_t) (0x01 << 2))

  UART4 Receiver enable [0] (in _UART4_CR2)
- #define _UART4_TEN ((uint8_t) (0x01 << 3))

  UART4 Transmitter enable [0] (in _UART4_CR2)
- #define _UART4_ILIEN ((uint8_t) (0x01 << 4))

  UART4 IDLE Line interrupt enable [0] (in _UART4_CR2)
- #define _UART4_RIEN ((uint8_t) (0x01 << 5))

  UART4 Receiver interrupt enable [0] (in _UART4_CR2)
- #define _UART4_TCIEN ((uint8_t) (0x01 << 6))

  UART4 Transmission complete interrupt enable [0] (in _UART4_CR2)
- #define _UART4_TIEN ((uint8_t) (0x01 << 7))

  UART4 Transmitter interrupt enable [0] (in _UART4_CR2)
- #define _UART4_LBCL ((uint8_t) (0x01 << 0))

  UART4 Last bit clock pulse [0] (in _UART4_CR3)
- #define _UART4_CPHA ((uint8_t) (0x01 << 1))

  UART4 Clock phase [0] (in _UART4_CR3)
- #define _UART4_CPOL ((uint8_t) (0x01 << 2))

  UART4 Clock polarity [0] (in _UART4_CR3)
- #define _UART4_CKEN ((uint8_t) (0x01 << 3))

  UART4 Clock enable [0] (in _UART4_CR3)
- #define _UART4_STOP ((uint8_t) (0x03 << 4))

  UART4 STOP bits [1:0] (in _UART4_CR3)
- #define _UART4_STOP0 ((uint8_t) (0x01 << 4))

  UART4 STOP bits [0] (in _UART4_CR3)
- #define _UART4_STOP1 ((uint8_t) (0x01 << 5))

  UART4 STOP bits [1] (in _UART4_CR3)
- #define _UART4_LINEN ((uint8_t) (0x01 << 6))

  UART4 LIN mode enable [0] (in _UART4_CR3)
- #define _UART4_ADD ((uint8_t) (0x0F << 0))

  UART4 Address of the UART node [3:0] (in _UART4_CR4)
- #define _UART4_ADD0 ((uint8_t) (0x01 << 0))

  UART4 Address of the UART node [0] (in _UART4_CR4)
- #define _UART4_ADD1 ((uint8_t) (0x01 << 1))

  UART4 Address of the UART node [1] (in _UART4_CR4)
- #define _UART4_ADD2 ((uint8_t) (0x01 << 2))

  UART4 Address of the UART node [2] (in _UART4_CR4)
- #define _UART4_ADD3 ((uint8_t) (0x01 << 3))

  UART4 Address of the UART node [3] (in _UART4_CR4)

- #define _UART4_LBDF ((uint8_t) (0x01 << 4))
  UART4 LIN Break Detection Flag [0] (in _UART4_CR4)
- #define _UART4_LBDL ((uint8_t) (0x01 << 5))
  UART4 LIN Break Detection Length [0] (in _UART4_CR4)
- #define _UART4_LBDIEN ((uint8_t) (0x01 << 6))
  UART4 LIN Break Detection Interrupt Enable [0] (in _UART4_CR4)
- #define _UART4_IREN ((uint8_t) (0x01 << 1))
  UART4 IrDA mode Enable [0] (in _UART4_CR5)
- #define _UART4_IRLP ((uint8_t) (0x01 << 2))
  UART4 IrDA Low Power [0] (in _UART4_CR5)
- #define _UART4_HDSEL ((uint8_t) (0x01 << 3))
  UART4 Half-Duplex Selection [0] (in _UART4_CR5)
- #define _UART4_NACK ((uint8_t) (0x01 << 4))
  UART4 Smartcard NACK enable [0] (in _UART4_CR5)
- #define _UART4_SCEN ((uint8_t) (0x01 << 5))
  UART4 Smartcard mode enable [0] (in _UART4_CR5)
- #define _UART4_LSF ((uint8_t) (0x01 << 0))
  UART4 LIN Sync Field [0] (in _UART4_CR6)
- #define _UART4_LHDF ((uint8_t) (0x01 << 1))
  UART4 LIN Header Detection Flag [0] (in _UART4_CR6)
- #define _UART4_LHDIEN ((uint8_t) (0x01 << 2))
  UART4 LIN Header Detection Interrupt Enable [0] (in _UART4_CR6)
- #define _UART4_LASE ((uint8_t) (0x01 << 4))
  UART4 LIN automatic resynchronisation enable [0] (in _UART4_CR6)
- #define _UART4_LSLV ((uint8_t) (0x01 << 5))
  UART4 LIN Slave Enable [0] (in _UART4_CR6)
- #define _UART4_LDUM ((uint8_t) (0x01 << 7))
  UART4 LIN Divider Update Method [0] (in _UART4_CR6)
- #define _TIM1 _SFR(TIM1_t, TIM1_AddressBase)
  TIM1 struct/bit access.
- #define _TIM1_CR1 _SFR(uint8_t, TIM1_AddressBase+0x00)
  TIM1 control register 1.
- #define _TIM1_CR2 _SFR(uint8_t, TIM1_AddressBase+0x01)
  TIM1 control register 2.
- #define _TIM1_SMCR _SFR(uint8_t, TIM1_AddressBase+0x02)
  TIM1 Slave mode control register.
- #define _TIM1_ETR _SFR(uint8_t, TIM1_AddressBase+0x03)
  TIM1 External trigger register.
- #define _TIM1_IER _SFR(uint8_t, TIM1_AddressBase+0x04)
  TIM1 interrupt enable register.
- #define _TIM1_SR1 _SFR(uint8_t, TIM1_AddressBase+0x05)
  TIM1 status register 1.
- #define _TIM1_SR2 _SFR(uint8_t, TIM1_AddressBase+0x06)
  TIM1 status register 2.
- #define _TIM1_EGR _SFR(uint8_t, TIM1_AddressBase+0x07)
  TIM1 Event generation register.
- #define _TIM1_CCMR1 _SFR(uint8_t, TIM1_AddressBase+0x08)
  TIM1 Capture/compare mode register 1.
- #define _TIM1_CCMR2 _SFR(uint8_t, TIM1_AddressBase+0x09)
  TIM1 Capture/compare mode register 2.
- #define _TIM1_CCMR3 _SFR(uint8_t, TIM1_AddressBase+0x0A)

TIM1 Capture/compare mode register 3.

- #define _TIM1_CCMR4 _SFR(uint8_t, TIM1_AddressBase+0x0B)

  TIM1 Capture/compare mode register 4.
- #define _TIM1_CCER1 _SFR(uint8_t, TIM1_AddressBase+0x0C)

  TIM1 Capture/compare enable register 1.
- #define _TIM1_CCER2 _SFR(uint8_t, TIM1_AddressBase+0x0D)

  TIM1 Capture/compare enable register 2.
- #define _TIM1_CNTRH _SFR(uint8_t, TIM1_AddressBase+0x0E)

  TIM1 counter register high byte.
- #define _TIM1_CNTRL _SFR(uint8_t, TIM1_AddressBase+0x0F)

  TIM1 counter register low byte.
- #define _TIM1_PSCRH _SFR(uint8_t, TIM1_AddressBase+0x10)

  TIM1 clock prescaler register high byte.
- #define _TIM1_PSCRL _SFR(uint8_t, TIM1_AddressBase+0x11)

  TIM1 clock prescaler register low byte.
- #define _TIM1_ARRH _SFR(uint8_t, TIM1_AddressBase+0x12)

  TIM1 auto-reload register high byte.
- #define _TIM1_ARRL _SFR(uint8_t, TIM1_AddressBase+0x13)

  TIM1 auto-reload register low byte.
- #define _TIM1_RCR _SFR(uint8_t, TIM1_AddressBase+0x14)

  TIM1 Repetition counter.
- #define _TIM1_CCR1H _SFR(uint8_t, TIM1_AddressBase+0x15)

  TIM1 16-bit capture/compare value 1 high byte.
- #define _TIM1_CCR1L _SFR(uint8_t, TIM1_AddressBase+0x16)

  TIM1 16-bit capture/compare value 1 low byte.
- #define _TIM1_CCR2H _SFR(uint8_t, TIM1_AddressBase+0x17)

  TIM1 16-bit capture/compare value 2 high byte.
- #define _TIM1_CCR2L _SFR(uint8_t, TIM1_AddressBase+0x18)

  TIM1 16-bit capture/compare value 2 low byte.
- #define _TIM1_CCR3H _SFR(uint8_t, TIM1_AddressBase+0x19)

  TIM1 16-bit capture/compare value 3 high byte.
- #define _TIM1_CCR3L _SFR(uint8_t, TIM1_AddressBase+0x1A)

  TIM1 16-bit capture/compare value 3 low byte.
- #define _TIM1_CCR4H _SFR(uint8_t, TIM1_AddressBase+0x1B)

  TIM1 16-bit capture/compare value 4 high byte.
- #define _TIM1_CCR4L _SFR(uint8_t, TIM1_AddressBase+0x1C)

  TIM1 16-bit capture/compare value 4 low byte.
- #define _TIM1_BKR _SFR(uint8_t, TIM1_AddressBase+0x1D)

  TIM1 Break register.
- #define _TIM1_DTR _SFR(uint8_t, TIM1_AddressBase+0x1E)

  TIM1 Dead-time register.
- #define _TIM1_OISR _SFR(uint8_t, TIM1_AddressBase+0x1F)

  TIM1 Output idle state register.
- #define _TIM1_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM1 control register 1 reset value.
- #define _TIM1_CR2_RESET_VALUE ((uint8_t) 0x00)

  TIM1 control register 2 reset value.
- #define _TIM1_SMCR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Slave mode control register reset value.
- #define _TIM1_ETR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 External trigger register reset value.

- #define _TIM1_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM1 interrupt enable register reset value.
- #define _TIM1_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM1 status register 1 reset value.
- #define _TIM1_SR2_RESET_VALUE ((uint8_t) 0x00)

  TIM1 status register 2 reset value.
- #define _TIM1_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Event generation register reset value.
- #define _TIM1_CCMR1_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Capture/compare mode register 1 reset value.
- #define _TIM1_CCMR2_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Capture/compare mode register 2 reset value.
- #define _TIM1_CCMR3_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Capture/compare mode register 3 reset value.
- #define _TIM1_CCMR4_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Capture/compare mode register 4 reset value.
- #define _TIM1_CCER1_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Capture/compare enable register 1 reset value.
- #define _TIM1_CCER2_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Capture/compare enable register 2 reset value.
- #define _TIM1_CNTRH_RESET_VALUE ((uint8_t) 0x00)

  TIM1 counter register high byte reset value.
- #define _TIM1_CNTRL_RESET_VALUE ((uint8_t) 0x00)

  TIM1 counter register low byte reset value.
- #define _TIM1_PSCRH_RESET_VALUE ((uint8_t) 0x00)

  TIM1 clock prescaler register high byte reset value.
- #define _TIM1_PSCRL_RESET_VALUE ((uint8_t) 0x00)

  TIM1 clock prescaler register low byte reset value.
- #define _TIM1_ARRH_RESET_VALUE ((uint8_t) 0xFF)

  TIM1 auto-reload register high byte reset value.
- #define _TIM1_ARRL_RESET_VALUE ((uint8_t) 0xFF)

  TIM1 auto-reload register low byte reset value.
- #define _TIM1_RCR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Repetition counter reset value.
- #define _TIM1_CCR1H_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 1 high byte reset value.
- #define _TIM1_CCR1L_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 1 low byte reset value.
- #define _TIM1_CCR2H_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 2 high byte reset value.
- #define _TIM1_CCR2L_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 2 low byte reset value.
- #define _TIM1_CCR3H_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 3 high byte reset value.
- #define _TIM1_CCR3L_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 3 low byte reset value.
- #define _TIM1_CCR4H_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 4 high byte reset value.
- #define _TIM1_CCR4L_RESET_VALUE ((uint8_t) 0x00)

  TIM1 16-bit capture/compare value 4 low byte reset value.
- #define _TIM1_BKR_RESET_VALUE ((uint8_t) 0x00)

TIM1 Break register reset value.

- #define _TIM1_DTR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Dead-time register reset value.
- #define _TIM1_OISR_RESET_VALUE ((uint8_t) 0x00)

  TIM1 Output idle state register reset value.
- #define _TIM1_CEN ((uint8_t) (0x01 << 0))

  TIM1 Counter enable [0] (in _TIM1_CR1)
- #define _TIM1_UDIS ((uint8_t) (0x01 << 1))

  TIM1 Update disable [0] (in _TIM1_CR1)
- #define _TIM1_URS ((uint8_t) (0x01 << 2))

  TIM1 Update request source [0] (in _TIM1_CR1)
- #define _TIM1_OPM ((uint8_t) (0x01 << 3))

  TIM1 One-pulse mode [0] (in _TIM1_CR1)
- #define _TIM1_DIR ((uint8_t) (0x01 << 4))

  TIM1 Direction [0] (in _TIM1_CR1)
- #define _TIM1_CMS ((uint8_t) (0x03 << 5))

  TIM1 Center-aligned mode selection [1:0] (in _TIM1_CR1)
- #define _TIM1_CMS0 ((uint8_t) (0x01 << 5))

  TIM1 Center-aligned mode selection [0] (in _TIM1_CR1)
- #define _TIM1_CMS1 ((uint8_t) (0x01 << 6))

  TIM1 Center-aligned mode selection [1] (in _TIM1_CR1)
- #define _TIM1_ARPE ((uint8_t) (0x01 << 7))

  TIM1 Auto-reload preload enable [0] (in _TIM1_CR1)
- #define _TIM1_CCPC ((uint8_t) (0x01 << 0))

  TIM1 Capture/compare preloaded control [0] (in _TIM1_CR2)
- #define _TIM1_COMS ((uint8_t) (0x01 << 2))

  TIM1 Capture/compare control update selection [0] (in _TIM1_CR2)
- #define _TIM1_MMS ((uint8_t) (0x07 << 4))

  TIM1 Master mode selection [2:0] (in _TIM1_CR2)
- #define _TIM1_MMS0 ((uint8_t) (0x01 << 4))

  TIM1 Master mode selection [0] (in _TIM1_CR2)
- #define _TIM1_MMS1 ((uint8_t) (0x01 << 5))

  TIM1 Master mode selection [1] (in _TIM1_CR2)
- #define _TIM1_MMS2 ((uint8_t) (0x01 << 6))

  TIM1 Master mode selection [2] (in _TIM1_CR2)
- #define _TIM1_SMS ((uint8_t) (0x07 << 0))

  TIM1 Clock/trigger/slave mode selection [2:0] (in _TIM1_SMCR)
- #define _TIM1_SMS0 ((uint8_t) (0x01 << 0))

  TIM1 Clock/trigger/slave mode selection [0] (in _TIM1_SMCR)
- #define _TIM1_SMS1 ((uint8_t) (0x01 << 1))

  TIM1 Clock/trigger/slave mode selection [1] (in _TIM1_SMCR)
- #define _TIM1_SMS2 ((uint8_t) (0x01 << 2))

  TIM1 Clock/trigger/slave mode selection [2] (in _TIM1_SMCR)
- #define _TIM1_TS ((uint8_t) (0x07 << 4))

  TIM1 Trigger selection [2:0] (in _TIM1_SMCR)
- #define _TIM1_TS0 ((uint8_t) (0x01 << 4))

  TIM1 Trigger selection [0] (in _TIM1_SMCR)
- #define _TIM1_TS1 ((uint8_t) (0x01 << 5))

  TIM1 Trigger selection [1] (in _TIM1_SMCR)
- #define _TIM1_TS2 ((uint8_t) (0x01 << 6))

  TIM1 Trigger selection [2] (in _TIM1_SMCR)

- #define _TIM1_MSM ((uint8_t) (0x01 << 7))
  TIM1 Master/slave mode [0] (in _TIM1_SMCR)
- #define _TIM1_ETF ((uint8_t) (0x0F << 0))
  TIM1 External trigger filter [3:0] (in _TIM1_ETR)
- #define _TIM1_ETF0 ((uint8_t) (0x01 << 0))
  TIM1 External trigger filter [0] (in _TIM1_ETR)
- #define _TIM1_ETF1 ((uint8_t) (0x01 << 1))
  TIM1 External trigger filter [1] (in _TIM1_ETR)
- #define _TIM1_ETF2 ((uint8_t) (0x01 << 2))
  TIM1 External trigger filter [2] (in _TIM1_ETR)
- #define _TIM1_ETF3 ((uint8_t) (0x01 << 3))
  TIM1 External trigger filter [3] (in _TIM1_ETR)
- #define _TIM1_ETPS ((uint8_t) (0x03 << 4))
  TIM1 External trigger prescaler [1:0] (in _TIM1_ETR)
- #define _TIM1_ETPS0 ((uint8_t) (0x01 << 4))
  TIM1 External trigger prescaler [0] (in _TIM1_ETR)
- #define _TIM1_ETPS1 ((uint8_t) (0x01 << 5))
  TIM1 External trigger prescaler [1] (in _TIM1_ETR)
- #define _TIM1_ECE ((uint8_t) (0x01 << 6))
  TIM1 External clock enable [0] (in _TIM1_ETR)
- #define _TIM1_ETP ((uint8_t) (0x01 << 7))
  TIM1 External trigger polarity [0] (in _TIM1_ETR)
- #define _TIM1_UIE ((uint8_t) (0x01 << 0))
  TIM1 Update interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_CC1IE ((uint8_t) (0x01 << 1))
  TIM1 Capture/compare 1 interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_CC2IE ((uint8_t) (0x01 << 2))
  TIM1 Capture/compare 2 interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_CC3IE ((uint8_t) (0x01 << 3))
  TIM1 Capture/compare 3 interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_CC4IE ((uint8_t) (0x01 << 4))
  TIM1 Capture/compare 4 interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_COMIE ((uint8_t) (0x01 << 5))
  TIM1 Commutation interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_TIE ((uint8_t) (0x01 << 6))
  TIM1 Trigger interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_BIE ((uint8_t) (0x01 << 7))
  TIM1 Break interrupt enable [0] (in _TIM1_IER)
- #define _TIM1_UIF ((uint8_t) (0x01 << 0))
  TIM1 Update interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_CC1IF ((uint8_t) (0x01 << 1))
  TIM1 Capture/compare 1 interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_CC2IF ((uint8_t) (0x01 << 2))
  TIM1 Capture/compare 2 interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_CC3IF ((uint8_t) (0x01 << 3))
  TIM1 Capture/compare 3 interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_CC4IF ((uint8_t) (0x01 << 4))
  TIM1 Capture/compare 4 interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_COMIF ((uint8_t) (0x01 << 5))
  TIM1 Commutation interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_TIF ((uint8_t) (0x01 << 6))

TIM1 Trigger interrupt flag [0] (in _TIM1_SR1)

- #define _TIM1_BIF ((uint8_t) (0x01 << 7))

  TIM1 Break interrupt flag [0] (in _TIM1_SR1)
- #define _TIM1_CC1OF ((uint8_t) (0x01 << 1))

  TIM1 Capture/compare 1 overcapture flag [0] (in _TIM1_SR2)
- #define _TIM1_CC2OF ((uint8_t) (0x01 << 2))

  TIM1 Capture/compare 2 overcapture flag [0] (in _TIM1_SR2)
- #define _TIM1_CC3OF ((uint8_t) (0x01 << 3))

  TIM1 Capture/compare 3 overcapture flag [0] (in _TIM1_SR2)
- #define _TIM1_CC4OF ((uint8_t) (0x01 << 4))

  TIM1 Capture/compare 4 overcapture flag [0] (in _TIM1_SR2)
- #define _TIM1_UG ((uint8_t) (0x01 << 0))

  TIM1 Update generation [0] (in _TIM1_EGR)
- #define _TIM1_CC1G ((uint8_t) (0x01 << 1))

  TIM1 Capture/compare 1 generation [0] (in _TIM1_EGR)
- #define _TIM1_CC2G ((uint8_t) (0x01 << 2))

  TIM1 Capture/compare 2 generation [0] (in _TIM1_EGR)
- #define _TIM1_CC3G ((uint8_t) (0x01 << 3))

  TIM1 Capture/compare 3 generation [0] (in _TIM1_EGR)
- #define _TIM1_CC4G ((uint8_t) (0x01 << 4))

  TIM1 Capture/compare 4 generation [0] (in _TIM1_EGR)
- #define _TIM1_COMG ((uint8_t) (0x01 << 5))

  TIM1 Capture/compare control update generation [0] (in _TIM1_EGR)
- #define _TIM1_TG ((uint8_t) (0x01 << 6))

  TIM1 Trigger generation [0] (in _TIM1_EGR)
- #define _TIM1_BG ((uint8_t) (0x01 << 7))

  TIM1 Break generation [0] (in _TIM1_EGR)
- #define _TIM1_CC1S ((uint8_t) (0x03 << 0))

  TIM1 Compare 1 selection [1:0] (in _TIM1_CCMR1)
- #define _TIM1_CC1S0 ((uint8_t) (0x01 << 0))

  TIM1 Compare 1 selection [0] (in _TIM1_CCMR1)
- #define _TIM1_CC1S1 ((uint8_t) (0x01 << 1))

  TIM1 Compare 1 selection [1] (in _TIM1_CCMR1)
- #define _TIM1_OC1FE ((uint8_t) (0x01 << 2))

  TIM1 Output compare 1 fast enable [0] (in _TIM1_CCMR1)
- #define _TIM1_OC1PE ((uint8_t) (0x01 << 3))

  TIM1 Output compare 1 preload enable [0] (in _TIM1_CCMR1)
- #define _TIM1_OC1M ((uint8_t) (0x07 << 4))

  TIM1 Output compare 1 mode [2:0] (in _TIM1_CCMR1)
- #define _TIM1_OC1M0 ((uint8_t) (0x01 << 4))

  TIM1 Output compare 1 mode [0] (in _TIM1_CCMR1)
- #define _TIM1_OC1M1 ((uint8_t) (0x01 << 5))

  TIM1 Output compare 1 mode [1] (in _TIM1_CCMR1)
- #define _TIM1_OC1M2 ((uint8_t) (0x01 << 6))

  TIM1 Output compare 1 mode [2] (in _TIM1_CCMR1)
- #define _TIM1_OC1CE ((uint8_t) (0x01 << 7))

  TIM1 Output compare 1 clear enable [0] (in _TIM1_CCMR1)
- #define _TIM1_IC1PSC ((uint8_t) (0x03 << 2))

  TIM1 Input capture 1 prescaler [1:0] (in _TIM1_CCMR1)
- #define _TIM1_IC1PSC0 ((uint8_t) (0x01 << 2))

  TIM1 Input capture 1 prescaler [0] (in _TIM1_CCMR1)

- #define _TIM1_IC1PSC1 ((uint8_t) (0x01 << 3))
  TIM1 Input capture 1 prescaler [1] (in _TIM1_CCMR1)
- #define _TIM1_IC1F ((uint8_t) (0x0F << 4))
  TIM1 Output compare 1 mode [3:0] (in _TIM1_CCMR1)
- #define _TIM1_IC1F0 ((uint8_t) (0x01 << 4))
  TIM1 Input capture 1 filter [0] (in _TIM1_CCMR1)
- #define _TIM1_IC1F1 ((uint8_t) (0x01 << 5))
  TIM1 Input capture 1 filter [1] (in _TIM1_CCMR1)
- #define _TIM1_IC1F2 ((uint8_t) (0x01 << 6))
  TIM1 Input capture 1 filter [2] (in _TIM1_CCMR1)
- #define _TIM1_IC1F3 ((uint8_t) (0x01 << 7))
  TIM1 Input capture 1 filter [3] (in _TIM1_CCMR1)
- #define _TIM1_CC2S ((uint8_t) (0x03 << 0))
  TIM1 Compare 2 selection [1:0] (in _TIM1_CCMR2)
- #define _TIM1_CC2S0 ((uint8_t) (0x01 << 0))
  TIM1 Compare 2 selection [0] (in _TIM1_CCMR2)
- #define _TIM1_CC2S1 ((uint8_t) (0x01 << 1))
  TIM1 Compare 2 selection [1] (in _TIM1_CCMR2)
- #define _TIM1_OC2FE ((uint8_t) (0x01 << 2))
  TIM1 Output compare 2 fast enable [0] (in _TIM1_CCMR2)
- #define _TIM1_OC2PE ((uint8_t) (0x01 << 3))
  TIM1 Output compare 2 preload enable [0] (in _TIM1_CCMR2)
- #define _TIM1_OC2M ((uint8_t) (0x07 << 4))
  TIM1 Output compare 2 mode [2:0] (in _TIM1_CCMR2)
- #define _TIM1_OC2M0 ((uint8_t) (0x01 << 4))
  TIM1 Output compare 2 mode [0] (in _TIM1_CCMR2)
- #define _TIM1_OC2M1 ((uint8_t) (0x01 << 5))
  TIM1 Output compare 2 mode [1] (in _TIM1_CCMR2)
- #define _TIM1_OC2M2 ((uint8_t) (0x01 << 6))
  TIM1 Output compare 2 mode [2] (in _TIM1_CCMR2)
- #define _TIM1_OC2CE ((uint8_t) (0x01 << 7))
  TIM1 Output compare 2 clear enable [0] (in _TIM1_CCMR2)
- #define _TIM1_IC2PSC ((uint8_t) (0x03 << 2))
  TIM1 Input capture 2 prescaler [1:0] (in _TIM1_CCMR2)
- #define _TIM1_IC2PSC0 ((uint8_t) (0x01 << 2))
  TIM1 Input capture 2 prescaler [0] (in _TIM1_CCMR2)
- #define _TIM1_IC2PSC1 ((uint8_t) (0x01 << 3))
  TIM1 Input capture 2 prescaler [1] (in _TIM1_CCMR2)
- #define _TIM1_IC2F ((uint8_t) (0x0F << 4))
  TIM1 Output compare 2 mode [3:0] (in _TIM1_CCMR2)
- #define _TIM1_IC2F0 ((uint8_t) (0x01 << 4))
  TIM1 Input capture 2 filter [0] (in _TIM1_CCMR2)
- #define _TIM1_IC2F1 ((uint8_t) (0x01 << 5))
  TIM1 Input capture 2 filter [1] (in _TIM1_CCMR2)
- #define _TIM1_IC2F2 ((uint8_t) (0x01 << 6))
  TIM1 Input capture 2 filter [2] (in _TIM1_CCMR2)
- #define _TIM1_IC2F3 ((uint8_t) (0x01 << 7))
  TIM1 Input capture 2 filter [3] (in _TIM1_CCMR2)
- #define _TIM1_CC3S ((uint8_t) (0x03 << 0))
  TIM1 Compare 3 selection [1:0] (in _TIM1_CCMR3)
- #define _TIM1_CC3S0 ((uint8_t) (0x01 << 0))

TIM1 Compare 3 selection [0] (in _TIM1_CCMR3)

- #define _TIM1_CC3S1 ((uint8_t) (0x01 << 1))

  TIM1 Compare 3 selection [1] (in _TIM1_CCMR3)
- #define _TIM1_OC3FE ((uint8_t) (0x01 << 2))

  TIM1 Output compare 3 fast enable [0] (in _TIM1_CCMR3)
- #define _TIM1_OC3PE ((uint8_t) (0x01 << 3))

  TIM1 Output compare 3 preload enable [0] (in _TIM1_CCMR3)
- #define _TIM1_OC3M ((uint8_t) (0x07 << 4))

  TIM1 Output compare 3 mode [2:0] (in _TIM1_CCMR3)
- #define _TIM1_OC3M0 ((uint8_t) (0x01 << 4))

  TIM1 Output compare 3 mode [0] (in _TIM1_CCMR3)
- #define _TIM1_OC3M1 ((uint8_t) (0x01 << 5))

  TIM1 Output compare 3 mode [1] (in _TIM1_CCMR3)
- #define _TIM1_OC3M2 ((uint8_t) (0x01 << 6))

  TIM1 Output compare 3 mode [2] (in _TIM1_CCMR3)
- #define _TIM1_OC3CE ((uint8_t) (0x01 << 7))

  TIM1 Output compare 3 clear enable [0] (in _TIM1_CCMR3)
- #define _TIM1_IC3PSC ((uint8_t) (0x03 << 2))

  TIM1 Input capture 3 prescaler [1:0] (in _TIM1_CCMR3)
- #define _TIM1_IC3PSC0 ((uint8_t) (0x01 << 2))

  TIM1 Input capture 3 prescaler [0] (in _TIM1_CCMR3)
- #define _TIM1_IC3PSC1 ((uint8_t) (0x01 << 3))

  TIM1 Input capture 3 prescaler [1] (in _TIM1_CCMR3)
- #define _TIM1_IC3F ((uint8_t) (0x0F << 4))

  TIM1 Output compare 3 mode [3:0] (in _TIM1_CCMR3)
- #define _TIM1_IC3F0 ((uint8_t) (0x01 << 4))

  TIM1 Input capture 3 filter [0] (in _TIM1_CCMR3)
- #define _TIM1_IC3F1 ((uint8_t) (0x01 << 5))

  TIM1 Input capture 3 filter [1] (in _TIM1_CCMR3)
- #define _TIM1_IC3F2 ((uint8_t) (0x01 << 6))

  TIM1 Input capture 3 filter [2] (in _TIM1_CCMR3)
- #define _TIM1_IC3F3 ((uint8_t) (0x01 << 7))

  TIM1 Input capture 3 filter [3] (in _TIM1_CCMR3)
- #define _TIM1_CC4S ((uint8_t) (0x03 << 0))

  TIM1 Compare 4 selection [1:0] (in _TIM1_CCMR4)
- #define _TIM1_CC4S0 ((uint8_t) (0x01 << 0))

  TIM1 Compare 4 selection [0] (in _TIM1_CCMR4)
- #define _TIM1_CC4S1 ((uint8_t) (0x01 << 1))

  TIM1 Compare 4 selection [1] (in _TIM1_CCMR4)
- #define _TIM1_OC4FE ((uint8_t) (0x01 << 2))

  TIM1 Output compare 4 fast enable [0] (in _TIM1_CCMR4)
- #define _TIM1_OC4PE ((uint8_t) (0x01 << 3))

  TIM1 Output compare 4 preload enable [0] (in _TIM1_CCMR4)
- #define _TIM1_OC4M ((uint8_t) (0x07 << 4))

  TIM1 Output compare 4 mode [2:0] (in _TIM1_CCMR4)
- #define _TIM1_OC4M0 ((uint8_t) (0x01 << 4))

  TIM1 Output compare 4 mode [0] (in _TIM1_CCMR4)
- #define _TIM1_OC4M1 ((uint8_t) (0x01 << 5))

  TIM1 Output compare 4 mode [1] (in _TIM1_CCMR4)
- #define _TIM1_OC4M2 ((uint8_t) (0x01 << 6))

  TIM1 Output compare 4 mode [2] (in _TIM1_CCMR4)

- #define _TIM1_OC4CE ((uint8_t) (0x01 << 7))
  TIM1 Output compare 4 clear enable [0] (in _TIM1_CCMR4)
- #define _TIM1_IC4PSC ((uint8_t) (0x03 << 2))
  TIM1 Input capture 4 prescaler [1:0] (in _TIM1_CCMR4)
- #define _TIM1_IC4PSC0 ((uint8_t) (0x01 << 2))
  TIM1 Input capture 4 prescaler [0] (in _TIM1_CCMR4)
- #define _TIM1_IC4PSC1 ((uint8_t) (0x01 << 3))
  TIM1 Input capture 4 prescaler [1] (in _TIM1_CCMR4)
- #define _TIM1_IC4F ((uint8_t) (0x0F << 4))
  TIM1 Output compare 4 mode [3:0] (in _TIM1_CCMR4)
- #define _TIM1_IC4F0 ((uint8_t) (0x01 << 4))
  TIM1 Input capture 4 filter [0] (in _TIM1_CCMR4)
- #define _TIM1_IC4F1 ((uint8_t) (0x01 << 5))
  TIM1 Input capture 4 filter [1] (in _TIM1_CCMR4)
- #define _TIM1_IC4F2 ((uint8_t) (0x01 << 6))
  TIM1 Input capture 4 filter [2] (in _TIM1_CCMR4)
- #define _TIM1_IC4F3 ((uint8_t) (0x01 << 7))
  TIM1 Input capture 4 filter [3] (in _TIM1_CCMR4)
- #define _TIM1_CC1E ((uint8_t) (0x01 << 0))
  TIM1 Capture/compare 1 output enable [0] (in _TIM1_CCER1)
- #define _TIM1_CC1P ((uint8_t) (0x01 << 1))
  TIM1 Capture/compare 1 output polarity [0] (in _TIM1_CCER1)
- #define _TIM1_CC1NE ((uint8_t) (0x01 << 2))
  TIM1 Capture/compare 1 complementary output enable [0] (in _TIM1_CCER1)
- #define _TIM1_CC1NP ((uint8_t) (0x01 << 3))
  TIM1 Capture/compare 1 complementary output polarity [0] (in _TIM1_CCER1)
- #define _TIM1_CC2E ((uint8_t) (0x01 << 4))
  TIM1 Capture/compare 2 output enable [0] (in _TIM1_CCER1)
- #define _TIM1_CC2P ((uint8_t) (0x01 << 5))
  TIM1 Capture/compare 2 output polarity [0] (in _TIM1_CCER1)
- #define _TIM1_CC2NE ((uint8_t) (0x01 << 6))
  TIM1 Capture/compare 2 complementary output enable [0] (in _TIM1_CCER1)
- #define _TIM1_CC2NP ((uint8_t) (0x01 << 7))
  TIM1 Capture/compare 2 complementary output polarity [0] (in _TIM1_CCER1)
- #define _TIM1_CC3E ((uint8_t) (0x01 << 0))
  TIM1 Capture/compare 3 output enable [0] (in _TIM1_CCER2)
- #define _TIM1_CC3P ((uint8_t) (0x01 << 1))
  TIM1 Capture/compare 3 output polarity [0] (in _TIM1_CCER2)
- #define _TIM1_CC3NE ((uint8_t) (0x01 << 2))
  TIM1 Capture/compare 3 complementary output enable [0] (in _TIM1_CCER2)
- #define _TIM1_CC3NP ((uint8_t) (0x01 << 3))
  TIM1 Capture/compare 3 complementary output polarity [0] (in _TIM1_CCER2)
- #define _TIM1_CC4E ((uint8_t) (0x01 << 4))
  TIM1 Capture/compare 4 output enable [0] (in _TIM1_CCER2)
- #define _TIM1_CC4P ((uint8_t) (0x01 << 5))
  TIM1 Capture/compare 4 output polarity [0] (in _TIM1_CCER2)
- #define _TIM1_LOCK ((int8_t) (0x03 << 0))
  TIM1 Lock configuration [1:0] (in _TIM1_BKR)
- #define _TIM1_LOCK0 ((uint8_t) (0x01 << 0))
  TIM1 Lock configuration [0] (in _TIM1_BKR)
- #define _TIM1_LOCK1 ((uint8_t) (0x01 << 1))

TIM1 Lock configuration [1] (in _TIM1_BKR)

- #define _TIM1_OSSI ((uint8_t) (0x01 << 2))

  TIM1 Off state selection for idle mode [0] (in _TIM1_BKR)
- #define _TIM1_OSSR ((uint8_t) (0x01 << 3))

  TIM1 Off state selection for Run mode [0] (in _TIM1_BKR)
- #define _TIM1_BKE ((uint8_t) (0x01 << 4))

  TIM1 Break enable [0] (in _TIM1_BKR)
- #define _TIM1_BKP ((uint8_t) (0x01 << 5))

  TIM1 Break polarity [0] (in _TIM1_BKR)
- #define _TIM1_AOE ((uint8_t) (0x01 << 6))

  TIM1 Automatic output enable [0] (in _TIM1_BKR)
- #define _TIM1_MOE ((uint8_t) (0x01 << 7))

  TIM1 Main output enable [0] (in _TIM1_BKR)
- #define _TIM1_OIS1 ((uint8_t) (0x01 << 0))

  TIM1 Output idle state 1 (OC1 output) [0] (in _TIM1_OISR)
- #define _TIM1_OIS1N ((uint8_t) (0x01 << 1))

  TIM1 Output idle state 1 (OC1N output) [0] (in _TIM1_OISR)
- #define _TIM1_OIS2 ((uint8_t) (0x01 << 2))

  TIM1 Output idle state 2 (OC2 output) [0] (in _TIM1_OISR)
- #define _TIM1_OIS2N ((uint8_t) (0x01 << 3))

  TIM1 Output idle state 2 (OC2N output) [0] (in _TIM1_OISR)
- #define _TIM1_OIS3 ((uint8_t) (0x01 << 4))

  TIM1 Output idle state 3 (OC3 output) [0] (in _TIM1_OISR)
- #define _TIM1_OIS3N ((uint8_t) (0x01 << 5))

  TIM1 Output idle state 3 (OC3N output) [0] (in _TIM1_OISR)
- #define _TIM1_OIS4 ((uint8_t) (0x01 << 6))

  TIM1 Output idle state 4 (OC4 output) [0] (in _TIM1_OISR)
- #define _TIM2 _SFR(TIM2_t, TIM2_AddressBase)

  TIM2 struct/bit access.
- #define _TIM2_CR1 _SFR(uint8_t, TIM2_AddressBase+0x00)

  TIM2 control register 1.
- #define _TIM2_IER _SFR(uint8_t, TIM2_AddressBase+0x01)

  TIM2 interrupt enable register.
- #define _TIM2_SR1 _SFR(uint8_t, TIM2_AddressBase+0x02)

  TIM2 status register 1.
- #define _TIM2_SR2 _SFR(uint8_t, TIM2_AddressBase+0x03)

  TIM2 status register 2.
- #define _TIM2_EGR _SFR(uint8_t, TIM2_AddressBase+0x04)

  TIM2 Event generation register.
- #define _TIM2_CCMR1 _SFR(uint8_t, TIM2_AddressBase+0x05)

  TIM2 Capture/compare mode register 1.
- #define _TIM2_CCMR2 _SFR(uint8_t, TIM2_AddressBase+0x06)

  TIM2 Capture/compare mode register 2.
- #define _TIM2_CCMR3 _SFR(uint8_t, TIM2_AddressBase+0x07)

  TIM2 Capture/compare mode register 3.
- #define _TIM2_CCER1 _SFR(uint8_t, TIM2_AddressBase+0x08)

  TIM2 Capture/compare enable register 1.
- #define _TIM2_CCER2 _SFR(uint8_t, TIM2_AddressBase+0x09)

  TIM2 Capture/compare enable register 2.
- #define _TIM2_CNTRH _SFR(uint8_t, TIM2_AddressBase+0x0A)

  TIM2 counter register high byte.

- #define _TIM2_CNTRL _SFR(uint8_t, TIM2_AddressBase+0x0B)

  TIM2 counter register low byte.
- #define _TIM2_PSCR _SFR(uint8_t, TIM2_AddressBase+0x0C)

  TIM2 clock prescaler register.
- #define _TIM2_ARRH _SFR(uint8_t, TIM2_AddressBase+0x0D)

  TIM2 auto-reload register high byte.
- #define _TIM2_ARRL _SFR(uint8_t, TIM2_AddressBase+0x0E)

  TIM2 auto-reload register low byte.
- #define _TIM2_CCR1H _SFR(uint8_t, TIM2_AddressBase+0x0F)

  TIM2 16-bit capture/compare value 1 high byte.
- #define _TIM2_CCR1L _SFR(uint8_t, TIM2_AddressBase+0x10)

  TIM2 16-bit capture/compare value 1 low byte.
- #define _TIM2_CCR2H _SFR(uint8_t, TIM2_AddressBase+0x11)

  TIM2 16-bit capture/compare value 2 high byte.
- #define _TIM2_CCR2L _SFR(uint8_t, TIM2_AddressBase+0x12)

  TIM2 16-bit capture/compare value 2 low byte.
- #define _TIM2_CCR3H _SFR(uint8_t, TIM2_AddressBase+0x13)

  TIM2 16-bit capture/compare value 3 high byte.
- #define _TIM2_CCR3L _SFR(uint8_t, TIM2_AddressBase+0x14)

  TIM2 16-bit capture/compare value 3 low byte.
- #define _TIM2_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 control register 1 reset value.
- #define _TIM2_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM2 interrupt enable register reset value.
- #define _TIM2_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 status register 1 reset value.
- #define _TIM2_SR2_RESET_VALUE ((uint8_t) 0x00)

  TIM2 status register 2 reset value.
- #define _TIM2_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Event generation register reset value.
- #define _TIM2_CCMR1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare mode register 1 reset value.
- #define _TIM2_CCMR2_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare mode register 2 reset value.
- #define _TIM2_CCMR3_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare mode register 3 reset value.
- #define _TIM2_CCER1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare enable register 1 reset value.
- #define _TIM2_CCER2_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare enable register 2 reset value.
- #define _TIM2_CNTRH_RESET_VALUE ((uint8_t) 0x00)

  TIM2 counter register high byte reset value.
- #define _TIM2_CNTRL_RESET_VALUE ((uint8_t) 0x00)

  TIM2 counter register low byte reset value.
- #define _TIM2_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 clock prescaler register reset value.
- #define _TIM2_ARRH_RESET_VALUE ((uint8_t) 0xFF)

  TIM2 auto-reload register high byte reset value.
- #define _TIM2_ARRL_RESET_VALUE ((uint8_t) 0xFF)

  TIM2 auto-reload register low byte reset value.
- #define _TIM2_CCR1H_RESET_VALUE ((uint8_t) 0x00)

TIM2 16-bit capture/compare value 1 high byte reset value.

- #define _TIM2_CCR1L_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 1 low byte reset value.
- #define _TIM2_CCR2H_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 2 high byte reset value.
- #define _TIM2_CCR2L_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 2 low byte reset value.
- #define _TIM2_CCR3H_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 3 high byte reset value.
- #define _TIM2_CCR3L_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 3 low byte reset value.
- #define _TIM2_CEN ((uint8_t) (0x01 << 0))

  TIM2 Counter enable [0] (in _TIM2_CR1)
- #define _TIM2_UDIS ((uint8_t) (0x01 << 1))

  TIM2 Update disable [0] (in _TIM2_CR1)
- #define _TIM2_URS ((uint8_t) (0x01 << 2))

  TIM2 Update request source [0] (in _TIM2_CR1)
- #define _TIM2_OPM ((uint8_t) (0x01 << 3))

  TIM2 One-pulse mode [0] (in _TIM2_CR1)
- #define _TIM2_ARPE ((uint8_t) (0x01 << 7))

  TIM2 Auto-reload preload enable [0] (in _TIM2_CR1)
- #define _TIM2_UIE ((uint8_t) (0x01 << 0))

  TIM2 Update interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_CC1IE ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_CC2IE ((uint8_t) (0x01 << 2))

  TIM2 Capture/compare 2 interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_CC3IE ((uint8_t) (0x01 << 3))

  TIM2 Capture/compare 3 interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_UIF ((uint8_t) (0x01 << 0))

  TIM2 Update interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC1IF ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC2IF ((uint8_t) (0x01 << 2))

  TIM2 Capture/compare 2 interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC3IF ((uint8_t) (0x01 << 3))

  TIM2 Capture/compare 3 interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC1OF ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 overcapture flag [0] (in _TIM2_SR2)
- #define _TIM2_CC2OF ((uint8_t) (0x01 << 2))

  TIM2 Capture/compare 2 overcapture flag [0] (in _TIM2_SR2)
- #define _TIM2_CC3OF ((uint8_t) (0x01 << 3))

  TIM2 Capture/compare 3 overcapture flag [0] (in _TIM2_SR2)
- #define _TIM2_UG ((uint8_t) (0x01 << 0))

  TIM2 Update generation [0] (in _TIM2_EGR)
- #define _TIM2_CC1G ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 generation [0] (in _TIM2_EGR)
- #define _TIM2_CC2G ((uint8_t) (0x01 << 2))

  TIM2 Capture/compare 2 generation [0] (in _TIM2_EGR)
- #define _TIM2_CC3G ((uint8_t) (0x01 << 3))

  TIM2 Capture/compare 3 generation [0] (in _TIM2_EGR)

- #define _TIM2_CC1S ((uint8_t) (0x03 << 0))

  TIM2 Compare 1 selection [1:0] (in _TIM2_CCMR1)
- #define _TIM2_CC1S0 ((uint8_t) (0x01 << 0))

  TIM2 Compare 1 selection [0] (in _TIM2_CCMR1)
- #define _TIM2_CC1S1 ((uint8_t) (0x01 << 1))

  TIM2 Compare 1 selection [1] (in _TIM2_CCMR1)
- #define _TIM2_OC1PE ((uint8_t) (0x01 << 3))

  TIM2 Output compare 1 preload enable [0] (in _TIM2_CCMR1)
- #define _TIM2_OC1M ((uint8_t) (0x07 << 4))

  TIM2 Output compare 1 mode [2:0] (in _TIM2_CCMR1)
- #define _TIM2_OC1M0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 1 mode [0] (in _TIM2_CCMR1)
- #define _TIM2_OC1M1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 1 mode [1] (in _TIM2_CCMR1)
- #define _TIM2_OC1M2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 1 mode [2] (in _TIM2_CCMR1)
- #define _TIM2_IC1PSC ((uint8_t) (0x03 << 2))

  TIM2 Input capture 1 prescaler [1:0] (in _TIM2_CCMR1)
- #define _TIM2_IC1PSC0 ((uint8_t) (0x01 << 2))

  TIM2 Input capture 1 prescaler [0] (in _TIM2_CCMR1)
- #define _TIM2_IC1PSC1 ((uint8_t) (0x01 << 3))

  TIM2 Input capture 1 prescaler [1] (in _TIM2_CCMR1)
- #define _TIM2_IC1F ((uint8_t) (0x0F << 4))

  TIM2 Output compare 1 mode [3:0] (in _TIM2_CCMR1)
- #define _TIM2_IC1F0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 1 mode [0] (in _TIM2_CCMR1)
- #define _TIM2_IC1F1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 1 mode [1] (in _TIM2_CCMR1)
- #define _TIM2_IC1F2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 1 mode [2] (in _TIM2_CCMR1)
- #define _TIM2_IC1F3 ((uint8_t) (0x01 << 7))

  TIM2 Output compare 1 mode [3] (in _TIM2_CCMR1)
- #define _TIM2_CC2S ((uint8_t) (0x03 << 0))

  TIM2 Compare 2 selection [1:0] (in _TIM2_CCMR2)
- #define _TIM2_CC2S0 ((uint8_t) (0x01 << 0))

  TIM2 Compare 2 selection [0] (in _TIM2_CCMR2)
- #define _TIM2_CC2S1 ((uint8_t) (0x01 << 1))

  TIM2 Compare 2 selection [1] (in _TIM2_CCMR2)
- #define _TIM2_OC2PE ((uint8_t) (0x01 << 3))

  TIM2 Output compare 2 preload enable [0] (in _TIM2_CCMR2)
- #define _TIM2_OC2M ((uint8_t) (0x07 << 4))

  TIM2 Output compare 2 mode [2:0] (in _TIM2_CCMR2)
- #define _TIM2_OC2M0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 2 mode [0] (in _TIM2_CCMR2)
- #define _TIM2_OC2M1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 2 mode [1] (in _TIM2_CCMR2)
- #define _TIM2_OC2M2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 2 mode [2] (in _TIM2_CCMR2)
- #define _TIM2_IC2PSC ((uint8_t) (0x03 << 2))

  TIM2 Input capture 2 prescaler [1:0] (in _TIM2_CCMR2)
- #define _TIM2_IC2PSC0 ((uint8_t) (0x01 << 2))

TIM2 Input capture 2 prescaler [0] (in _TIM2_CCMR2)

- #define _TIM2_IC2PSC1 ((uint8_t) (0x01 << 3))

  TIM2 Input capture 2 prescaler [1] (in _TIM2_CCMR2)
- #define _TIM2_IC2F ((uint8_t) (0x0F << 4))

  TIM2 Output compare 2 mode [3:0] (in _TIM2_CCMR2)
- #define _TIM2_IC2F0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 2 mode [0] (in _TIM2_CCMR2)
- #define _TIM2_IC2F1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 2 mode [1] (in _TIM2_CCMR2)
- #define _TIM2_IC2F2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 2 mode [2] (in _TIM2_CCMR2)
- #define _TIM2_IC2F3 ((uint8_t) (0x01 << 7))

  TIM2 Output compare 2 mode [3] (in _TIM2_CCMR2)
- #define _TIM2_CC3S ((uint8_t) (0x03 << 0))

  TIM2 Compare 3 selection [1:0] (in _TIM2_CCMR3)
- #define _TIM2_CC3S0 ((uint8_t) (0x01 << 0))

  TIM2 Compare 3 selection [0] (in _TIM2_CCMR3)
- #define _TIM2_CC3S1 ((uint8_t) (0x01 << 1))

  TIM2 Compare 3 selection [1] (in _TIM2_CCMR3)
- #define _TIM2_OC3PE ((uint8_t) (0x01 << 3))

  TIM2 Output compare 3 preload enable [0] (in _TIM2_CCMR3)
- #define _TIM2_OC3M ((uint8_t) (0x07 << 4))

  TIM2 Output compare 3 mode [2:0] (in _TIM2_CCMR3)
- #define _TIM2_OC3M0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 3 mode [0] (in _TIM2_CCMR3)
- #define _TIM2_OC3M1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 3 mode [1] (in _TIM2_CCMR3)
- #define _TIM2_OC3M2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 3 mode [2] (in _TIM2_CCMR3)
- #define _TIM2_IC3PSC ((uint8_t) (0x03 << 2))

  TIM2 Input capture 3 prescaler [1:0] (in _TIM2_CCMR3)
- #define _TIM2_IC3PSC0 ((uint8_t) (0x01 << 2))

  TIM2 Input capture 3 prescaler [0] (in _TIM2_CCMR3)
- #define _TIM2_IC3PSC1 ((uint8_t) (0x01 << 3))

  TIM2 Input capture 3 prescaler [1] (in _TIM2_CCMR3)
- #define _TIM2_IC3F ((uint8_t) (0x0F << 4))

  TIM2 Output compare 3 mode [3:0] (in _TIM2_CCMR3)
- #define _TIM2_IC3F0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 3 mode [0] (in _TIM2_CCMR3)
- #define _TIM2_IC3F1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 3 mode [1] (in _TIM2_CCMR3)
- #define _TIM2_IC3F2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 3 mode [2] (in _TIM2_CCMR3)
- #define _TIM2_IC3F3 ((uint8_t) (0x01 << 7))

  TIM2 Output compare 3 mode [3] (in _TIM2_CCMR3)
- #define _TIM2_CC1E ((uint8_t) (0x01 << 0))

  TIM2 Capture/compare 1 output enable [0] (in _TIM2_CCER1)
- #define _TIM2_CC1P ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 output polarity [0] (in _TIM2_CCER1)
- #define _TIM2_CC2E ((uint8_t) (0x01 << 4))

  TIM2 Capture/compare 2 output enable [0] (in _TIM2_CCER1)

- #define _TIM2_CC2P ((uint8_t) (0x01 << 5))
  TIM2 Capture/compare 2 output polarity [0] (in _TIM2_CCER1)
- #define _TIM2_CC3E ((uint8_t) (0x01 << 0))
  TIM2 Capture/compare 3 output enable [0] (in _TIM2_CCER2)
- #define _TIM2_CC3P ((uint8_t) (0x01 << 1))
  TIM2 Capture/compare 3 output polarity [0] (in _TIM2_CCER2)
- #define _TIM2_PSC ((uint8_t) (0x0F << 0))
  TIM2 prescaler [3:0] (in _TIM2_PSCR)
- #define _TIM2_PSC0 ((uint8_t) (0x01 << 0))
  TIM2 prescaler [0] (in _TIM2_PSCR)
- #define _TIM2_PSC1 ((uint8_t) (0x01 << 1))
  TIM2 prescaler [1] (in _TIM2_PSCR)
- #define _TIM2_PSC2 ((uint8_t) (0x01 << 2))
  TIM2 prescaler [2] (in _TIM2_PSCR)
- #define _TIM2_PSC3 ((uint8_t) (0x01 << 3))
  TIM2 prescaler [3] (in _TIM2_PSCR)
- #define _TIM3 _SFR(TIM3_t, TIM3_AddressBase)
  TIM3 struct/bit access.
- #define _TIM3_CR1 _SFR(uint8_t, TIM3_AddressBase+0x00)
  TIM3 control register 1.
- #define _TIM3_IER _SFR(uint8_t, TIM3_AddressBase+0x01)
  TIM3 interrupt enable register.
- #define _TIM3_SR1 _SFR(uint8_t, TIM3_AddressBase+0x02)
  TIM3 status register 1.
- #define _TIM3_SR2 _SFR(uint8_t, TIM3_AddressBase+0x03)
  TIM3 status register 2.
- #define _TIM3_EGR _SFR(uint8_t, TIM3_AddressBase+0x04)
  TIM3 Event generation register.
- #define _TIM3_CCMR1 _SFR(uint8_t, TIM3_AddressBase+0x05)
  TIM3 Capture/compare mode register 1.
- #define _TIM3_CCMR2 _SFR(uint8_t, TIM3_AddressBase+0x06)
  TIM3 Capture/compare mode register 2.
- #define _TIM3_CCER1 _SFR(uint8_t, TIM3_AddressBase+0x08)
  TIM3 Capture/compare enable register 1.
- #define _TIM3_CNTRH _SFR(uint8_t, TIM3_AddressBase+0x0A)
  TIM3 counter register high byte.
- #define _TIM3_CNTRL _SFR(uint8_t, TIM3_AddressBase+0x0B)
  TIM3 counter register low byte.
- #define _TIM3_PSCR _SFR(uint8_t, TIM3_AddressBase+0x0C)
  TIM3 clock prescaler register.
- #define _TIM3_ARRH _SFR(uint8_t, TIM3_AddressBase+0x0D)
  TIM3 auto-reload register high byte.
- #define _TIM3_ARRL _SFR(uint8_t, TIM3_AddressBase+0x0E)
  TIM3 auto-reload register low byte.
- #define _TIM3_CCR1H _SFR(uint8_t, TIM3_AddressBase+0x0F)
  TIM3 16-bit capture/compare value 1 high byte.
- #define _TIM3_CCR1L _SFR(uint8_t, TIM3_AddressBase+0x10)
  TIM3 16-bit capture/compare value 1 low byte.
- #define _TIM3_CCR2H _SFR(uint8_t, TIM3_AddressBase+0x11)
  TIM3 16-bit capture/compare value 2 high byte.
- #define _TIM3_CCR2L _SFR(uint8_t, TIM3_AddressBase+0x12)

TIM3 16-bit capture/compare value 2 low byte.

- #define _TIM3_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 control register 1 reset value.
- #define _TIM3_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM3 interrupt enable register reset value.
- #define _TIM3_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 status register 1 reset value.
- #define _TIM3_SR2_RESET_VALUE ((uint8_t) 0x00)

  TIM3 status register 2 reset value.
- #define _TIM3_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Event generation register reset value.
- #define _TIM3_CCMR1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Capture/compare mode register 1 reset value.
- #define _TIM3_CCMR2_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Capture/compare mode register 2 reset value.
- #define _TIM3_CCER1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Capture/compare enable register 1 reset value.
- #define _TIM3_CNTRH_RESET_VALUE ((uint8_t) 0x00)

  TIM3 counter register high byte reset value.
- #define _TIM3_CNTRL_RESET_VALUE ((uint8_t) 0x00)

  TIM3 counter register low byte reset value.
- #define _TIM3_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 clock prescaler register reset value.
- #define _TIM3_ARRH_RESET_VALUE ((uint8_t) 0xFF)

  TIM3 auto-reload register high byte reset value.
- #define _TIM3_ARRL_RESET_VALUE ((uint8_t) 0xFF)

  TIM3 auto-reload register low byte reset value.
- #define _TIM3_CCR1H_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 1 high byte reset value.
- #define _TIM3_CCR1L_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 1 low byte reset value.
- #define _TIM3_CCR2H_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 2 high byte reset value.
- #define _TIM3_CCR2L_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 2 low byte reset value.
- #define _TIM3_CEN ((uint8_t) (0x01 << 0))

  TIM3 Counter enable [0] (in _TIM3_CR1)
- #define _TIM3_UDIS ((uint8_t) (0x01 << 1))

  TIM3 Update disable [0] (in _TIM3_CR1)
- #define _TIM3_URS ((uint8_t) (0x01 << 2))

  TIM3 Update request source [0] (in _TIM3_CR1)
- #define _TIM3_OPM ((uint8_t) (0x01 << 3))

  TIM3 One-pulse mode [0] (in _TIM3_CR1)
- #define _TIM3_ARPE ((uint8_t) (0x01 << 7))

  TIM3 Auto-reload preload enable [0] (in _TIM3_CR1)
- #define _TIM3_UIE ((uint8_t) (0x01 << 0))

  TIM3 Update interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_CC1IE ((uint8_t) (0x01 << 1))

  TIM3 Capture/compare 1 interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_CC2IE ((uint8_t) (0x01 << 2))

  TIM3 Capture/compare 2 interrupt enable [0] (in _TIM3_IER)

- #define _TIM3_UIF ((uint8_t) (0x01 << 0))
  TIM3 Update interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_CC1IF ((uint8_t) (0x01 << 1))
  TIM3 Capture/compare 1 interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_CC2IF ((uint8_t) (0x01 << 2))
  TIM3 Capture/compare 2 interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_CC1OF ((uint8_t) (0x01 << 1))
  TIM3 Capture/compare 1 overcapture flag [0] (in _TIM3_SR2)
- #define _TIM3_CC2OF ((uint8_t) (0x01 << 2))
  TIM3 Capture/compare 2 overcapture flag [0] (in _TIM3_SR2)
- #define _TIM3_UG ((uint8_t) (0x01 << 0))
  TIM3 Update generation [0] (in _TIM3_EGR)
- #define _TIM3_CC1G ((uint8_t) (0x01 << 1))
  TIM3 Capture/compare 1 generation [0] (in _TIM3_EGR)
- #define _TIM3_CC2G ((uint8_t) (0x01 << 2))
  TIM3 Capture/compare 2 generation [0] (in _TIM3_EGR)
- #define _TIM3_CC1S ((uint8_t) (0x03 << 0))
  TIM3 Compare 1 selection [1:0] (in _TIM3_CCMR1)
- #define _TIM3_CC1S0 ((uint8_t) (0x01 << 0))
  TIM3 Compare 1 selection [0] (in _TIM3_CCMR1)
- #define _TIM3_CC1S1 ((uint8_t) (0x01 << 1))
  TIM3 Compare 1 selection [1] (in _TIM3_CCMR1)
- #define _TIM3_OC1PE ((uint8_t) (0x01 << 3))
  TIM3 Output compare 1 preload enable [0] (in _TIM3_CCMR1)
- #define _TIM3_OC1M ((uint8_t) (0x07 << 4))
  TIM3 Output compare 1 mode [2:0] (in _TIM3_CCMR1)
- #define _TIM3_OC1M0 ((uint8_t) (0x01 << 4))
  TIM3 Output compare 1 mode [0] (in _TIM3_CCMR1)
- #define _TIM3_OC1M1 ((uint8_t) (0x01 << 5))
  TIM3 Output compare 1 mode [1] (in _TIM3_CCMR1)
- #define _TIM3_OC1M2 ((uint8_t) (0x01 << 6))
  TIM3 Output compare 1 mode [2] (in _TIM3_CCMR1)
- #define _TIM3_IC1PSC ((uint8_t) (0x03 << 2))
  TIM3 Input capture 1 prescaler [1:0] (in _TIM3_CCMR1)
- #define _TIM3_IC1PSC0 ((uint8_t) (0x01 << 2))
  TIM3 Input capture 1 prescaler [0] (in _TIM3_CCMR1)
- #define _TIM3_IC1PSC1 ((uint8_t) (0x01 << 3))
  TIM3 Input capture 1 prescaler [1] (in _TIM3_CCMR1)
- #define _TIM3_IC1F ((uint8_t) (0x0F << 4))
  TIM3 Output compare 1 mode [3:0] (in _TIM3_CCMR1)
- #define _TIM3_IC1F0 ((uint8_t) (0x01 << 4))
  TIM3 Output compare 1 mode [0] (in _TIM3_CCMR1)
- #define _TIM3_IC1F1 ((uint8_t) (0x01 << 5))
  TIM3 Output compare 1 mode [1] (in _TIM3_CCMR1)
- #define _TIM3_IC1F2 ((uint8_t) (0x01 << 6))
  TIM3 Output compare 1 mode [2] (in _TIM3_CCMR1)
- #define _TIM3_IC1F3 ((uint8_t) (0x01 << 7))
  TIM3 Output compare 1 mode [3] (in _TIM3_CCMR1)
- #define _TIM3_CC2S ((uint8_t) (0x03 << 0))
  TIM3 Compare 2 selection [1:0] (in _TIM3_CCMR2)
- #define _TIM3_CC2S0 ((uint8_t) (0x01 << 0))

TIM3 Compare 2 selection [0] (in _TIM3_CCMR2)

- #define _TIM3_CC2S1 ((uint8_t) (0x01 << 1))

  TIM3 Compare 2 selection [1] (in _TIM3_CCMR2)
- #define _TIM3_OC2PE ((uint8_t) (0x01 << 3))

  TIM3 Output compare 2 preload enable [0] (in _TIM3_CCMR2)
- #define _TIM3_OC2M ((uint8_t) (0x07 << 4))

  TIM3 Output compare 2 mode [2:0] (in _TIM3_CCMR2)
- #define _TIM3_OC2M0 ((uint8_t) (0x01 << 4))

  TIM3 Output compare 2 mode [0] (in _TIM3_CCMR2)
- #define _TIM3_OC2M1 ((uint8_t) (0x01 << 5))

  TIM3 Output compare 2 mode [1] (in _TIM3_CCMR2)
- #define _TIM3_OC2M2 ((uint8_t) (0x01 << 6))

  TIM3 Output compare 2 mode [2] (in _TIM3_CCMR2)
- #define _TIM3_IC2PSC ((uint8_t) (0x03 << 2))

  TIM3 Input capture 2 prescaler [1:0] (in _TIM3_CCMR2)
- #define _TIM3_IC2PSC0 ((uint8_t) (0x01 << 2))

  TIM3 Input capture 2 prescaler [0] (in _TIM3_CCMR2)
- #define _TIM3_IC2PSC1 ((uint8_t) (0x01 << 3))

  TIM3 Input capture 2 prescaler [1] (in _TIM3_CCMR2)
- #define _TIM3_IC2F ((uint8_t) (0x0F << 4))

  TIM3 Output compare 2 mode [3:0] (in _TIM3_CCMR2)
- #define _TIM3_IC2F0 ((uint8_t) (0x01 << 4))

  TIM3 Output compare 2 mode [0] (in _TIM3_CCMR2)
- #define _TIM3_IC2F1 ((uint8_t) (0x01 << 5))

  TIM3 Output compare 2 mode [1] (in _TIM3_CCMR2)
- #define _TIM3_IC2F2 ((uint8_t) (0x01 << 6))

  TIM3 Output compare 2 mode [2] (in _TIM3_CCMR2)
- #define _TIM3_IC2F3 ((uint8_t) (0x01 << 7))

  TIM3 Output compare 2 mode [3] (in _TIM3_CCMR2)
- #define _TIM3_CC1E ((uint8_t) (0x01 << 0))

  TIM3 Capture/compare 1 output enable [0] (in _TIM3_CCER1)
- #define _TIM3_CC1P ((uint8_t) (0x01 << 1))

  TIM3 Capture/compare 1 output polarity [0] (in _TIM3_CCER1)
- #define _TIM3_CC2E ((uint8_t) (0x01 << 4))

  TIM3 Capture/compare 2 output enable [0] (in _TIM3_CCER1)
- #define _TIM3_CC2P ((uint8_t) (0x01 << 5))

  TIM3 Capture/compare 2 output polarity [0] (in _TIM3_CCER1)
- #define _TIM3_PSC ((uint8_t) (0x0F << 0))

  TIM3 clock prescaler [3:0] (in _TIM3_PSCR)
- #define _TIM3_PSC0 ((uint8_t) (0x01 << 0))

  TIM3 clock prescaler [0] (in _TIM3_PSCR)
- #define _TIM3_PSC1 ((uint8_t) (0x01 << 1))

  TIM3 clock prescaler [1] (in _TIM3_PSCR)
- #define _TIM3_PSC2 ((uint8_t) (0x01 << 2))

  TIM3 clock prescaler [2] (in _TIM3_PSCR)
- #define _TIM3_PSC3 ((uint8_t) (0x01 << 3))

  TIM3 clock prescaler [3] (in _TIM3_PSCR)
- #define _TIM4 _SFR(TIM4_t, TIM4_AddressBase)

  TIM4 struct/bit access.
- #define _TIM4_CR _SFR(uint8_t, TIM4_AddressBase+0x00)

  TIM4 control register.

- #define _TIM4_IER _SFR(uint8_t, TIM4_AddressBase+0x01)
  TIM4 interrupt enable register.
- #define _TIM4_SR _SFR(uint8_t, TIM4_AddressBase+0x02)
  TIM4 status register.
- #define _TIM4_EGR _SFR(uint8_t, TIM4_AddressBase+0x03)
  TIM4 event generation register.
- #define _TIM4_CNTR _SFR(uint8_t, TIM4_AddressBase+0x04)
  TIM4 counter register.
- #define _TIM4_PSCR _SFR(uint8_t, TIM4_AddressBase+0x05)
  TIM4 clock prescaler register.
- #define _TIM4_ARR _SFR(uint8_t, TIM4_AddressBase+0x06)
  TIM4 auto-reload register.
- #define _TIM4_CR_RESET_VALUE ((uint8_t) 0x00)
  TIM4 control register reset value.
- #define _TIM4_IER_RESET_VALUE ((uint8_t) 0x00)
  TIM4 interrupt enable register reset value.
- #define _TIM4_SR_RESET_VALUE ((uint8_t) 0x00)
  TIM4 status register reset value.
- #define _TIM4_EGR_RESET_VALUE ((uint8_t) 0x00)
  TIM4 event generation register reset value.
- #define _TIM4_CNTR_RESET_VALUE ((uint8_t) 0x00)
  TIM4 counter register reset value.
- #define _TIM4_PSCR_RESET_VALUE ((uint8_t) 0x00)
  TIM4 clock prescaler register reset value.
- #define _TIM4_ARR_RESET_VALUE ((uint8_t) 0xFF)
  TIM4 auto-reload register reset value.
- #define _TIM4_CEN ((uint8_t) (0x01 << 0))
  TIM4 Counter enable [0] (in _TIM4_CR)
- #define _TIM4_UDIS ((uint8_t) (0x01 << 1))
  TIM4 Update disable [0] (in _TIM4_CR)
- #define _TIM4_URS ((uint8_t) (0x01 << 2))
  TIM4 Update request source [0] (in _TIM4_CR)
- #define _TIM4_OPM ((uint8_t) (0x01 << 3))
  TIM4 One-pulse mode [0] (in _TIM4_CR)
- #define _TIM4_ARPE ((uint8_t) (0x01 << 7))
  TIM4 Auto-reload preload enable [0] (in _TIM4_CR)
- #define _TIM4_UIE ((uint8_t) (0x01 << 0))
  TIM4 Update interrupt enable [0] (in _TIM4_IER)
- #define _TIM4_UIF ((uint8_t) (0x01 << 0))
  TIM4 Update interrupt flag [0] (in _TIM4_SR)
- #define _TIM4_UG ((uint8_t) (0x01 << 0))
  TIM4 Update generation [0] (in _TIM4_EGR)
- #define _TIM4_PSC ((uint8_t) (0x07 << 0))
  TIM4 clock prescaler [2:0] (in _TIM4_PSCR)
- #define _TIM4_PSC0 ((uint8_t) (0x01 << 0))
  TIM4 clock prescaler [0] (in _TIM4_PSCR)
- #define _TIM4_PSC1 ((uint8_t) (0x01 << 1))
  TIM4 clock prescaler [1] (in _TIM4_PSCR)
- #define _TIM4_PSC2 ((uint8_t) (0x01 << 2))
  TIM4 clock prescaler [2] (in _TIM4_PSCR)
- #define _TIM5 _SFR(TIM5_t, TIM5_AddressBase)

TIM5 struct/bit access.

- #define _TIM5_CR1 _SFR(uint8_t, TIM5_AddressBase+0x00)

  TIM5 control register 1.
- #define _TIM5_CR2 _SFR(uint8_t, TIM5_AddressBase+0x01)

  TIM5 control register 2.
- #define _TIM5_SMCR _SFR(uint8_t, TIM5_AddressBase+0x02)

  TIM5 Slave mode control register.
- #define _TIM5_IER _SFR(uint8_t, TIM5_AddressBase+0x03)

  TIM5 interrupt enable register.
- #define _TIM5_SR1 _SFR(uint8_t, TIM5_AddressBase+0x04)

  TIM5 status register 1.
- #define _TIM5_SR2 _SFR(uint8_t, TIM5_AddressBase+0x05)

  TIM5 status register 2.
- #define _TIM5_EGR _SFR(uint8_t, TIM5_AddressBase+0x06)

  TIM5 Event generation register.
- #define _TIM5_CCMR1 _SFR(uint8_t, TIM5_AddressBase+0x07)

  TIM5 Capture/compare mode register 1.
- #define _TIM5_CCMR2 _SFR(uint8_t, TIM5_AddressBase+0x08)

  TIM5 Capture/compare mode register 2.
- #define _TIM5_CCMR3 _SFR(uint8_t, TIM5_AddressBase+0x09)

  TIM5 Capture/compare mode register 3.
- #define _TIM5_CCER1 _SFR(uint8_t, TIM5_AddressBase+0x0A)

  TIM5 Capture/compare enable register 1.
- #define _TIM5_CCER2 _SFR(uint8_t, TIM5_AddressBase+0x0B)

  TIM5 Capture/compare enable register 2.
- #define _TIM5_CNTRH _SFR(uint8_t, TIM5_AddressBase+0x0C)

  TIM5 counter register high byte.
- #define _TIM5_CNTRL _SFR(uint8_t, TIM5_AddressBase+0x0D)

  TIM5 counter register low byte.
- #define _TIM5_PSCR _SFR(uint8_t, TIM5_AddressBase+0x0E)

  TIM5 clock prescaler register.
- #define _TIM5_ARRH _SFR(uint8_t, TIM5_AddressBase+0x0F)

  TIM5 auto-reload register high byte.
- #define _TIM5_ARRL _SFR(uint8_t, TIM5_AddressBase+0x10)

  TIM5 auto-reload register low byte.
- #define _TIM5_CCR1H _SFR(uint8_t, TIM5_AddressBase+0x11)

  TIM5 16-bit capture/compare value 1 high byte.
- #define _TIM5_CCR1L _SFR(uint8_t, TIM5_AddressBase+0x12)

  TIM5 16-bit capture/compare value 1 low byte.
- #define _TIM5_CCR2H _SFR(uint8_t, TIM5_AddressBase+0x13)

  TIM5 16-bit capture/compare value 2 high byte.
- #define _TIM5_CCR2L _SFR(uint8_t, TIM5_AddressBase+0x14)

  TIM5 16-bit capture/compare value 2 low byte.
- #define _TIM5_CCR3H _SFR(uint8_t, TIM5_AddressBase+0x15)

  TIM5 16-bit capture/compare value 3 high byte.
- #define _TIM5_CCR3L _SFR(uint8_t, TIM5_AddressBase+0x16)

  TIM5 16-bit capture/compare value 3 low byte.
- #define _TIM5_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM5 control register 1 reset value.
- #define _TIM5_CR2_RESET_VALUE ((uint8_t) 0x00)

  TIM5 control register 2 reset value.

- #define _TIM5_SMCR_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Slave mode control register reset value.
- #define _TIM5_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM5 interrupt enable register reset value.
- #define _TIM5_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM5 status register 1 reset value.
- #define _TIM5_SR2_RESET_VALUE ((uint8_t) 0x00)

  TIM5 status register 2 reset value.
- #define _TIM5_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Event generation register reset value.
- #define _TIM5_CCMR1_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Capture/compare mode register 1 reset value.
- #define _TIM5_CCMR2_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Capture/compare mode register 2 reset value.
- #define _TIM5_CCMR3_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Capture/compare mode register 3 reset value.
- #define _TIM5_CCER1_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Capture/compare enable register 1 reset value.
- #define _TIM5_CCER2_RESET_VALUE ((uint8_t) 0x00)

  TIM5 Capture/compare enable register 2 reset value.
- #define _TIM5_CNTRH_RESET_VALUE ((uint8_t) 0x00)

  TIM5 counter register high byte reset value.
- #define _TIM5_CNTRL_RESET_VALUE ((uint8_t) 0x00)

  TIM5 counter register low byte reset value.
- #define _TIM5_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM5 clock prescaler register reset value.
- #define _TIM5_ARRH_RESET_VALUE ((uint8_t) 0xFF)

  TIM5 auto-reload register high byte reset value.
- #define _TIM5_ARRL_RESET_VALUE ((uint8_t) 0xFF)

  TIM5 auto-reload register low byte reset value.
- #define _TIM5_CCR1H_RESET_VALUE ((uint8_t) 0x00)

  TIM5 16-bit capture/compare value 1 high byte reset value.
- #define _TIM5_CCR1L_RESET_VALUE ((uint8_t) 0x00)

  TIM5 16-bit capture/compare value 1 low byte reset value.
- #define _TIM5_CCR2H_RESET_VALUE ((uint8_t) 0x00)

  TIM5 16-bit capture/compare value 2 high byte reset value.
- #define _TIM5_CCR2L_RESET_VALUE ((uint8_t) 0x00)

  TIM5 16-bit capture/compare value 2 low byte reset value.
- #define _TIM5_CCR3H_RESET_VALUE ((uint8_t) 0x00)

  TIM5 16-bit capture/compare value 3 high byte reset value.
- #define _TIM5_CCR3L_RESET_VALUE ((uint8_t) 0x00)

  TIM5 16-bit capture/compare value 3 low byte reset value.
- #define _TIM5_CEN ((uint8_t) (0x01 << 0))

  TIM5 Counter enable [0] (in _TIM5_CR1)
- #define _TIM5_UDIS ((uint8_t) (0x01 << 1))

  TIM5 Update disable [0] (in _TIM5_CR1)
- #define _TIM5_URS ((uint8_t) (0x01 << 2))

  TIM5 Update request source [0] (in _TIM5_CR1)
- #define _TIM5_OPM ((uint8_t) (0x01 << 3))

  TIM5 One-pulse mode [0] (in _TIM5_CR1)
- #define _TIM5_ARPE ((uint8_t) (0x01 << 7))

TIM5 Auto-reload preload enable [0] (in _TIM5_CR1)

- #define _TIM5_CCPC ((uint8_t) (0x01 << 0))

  TIM5 Capture/compare preloaded control [0] (in _TIM5_CR2)
- #define _TIM5_COMS ((uint8_t) (0x01 << 2))

  TIM5 Capture/compare control update selection [0] (in _TIM5_CR2)
- #define _TIM5_MMS ((uint8_t) (0x07 << 4))

  TIM5 Master mode selection [2:0] (in _TIM5_CR2)
- #define _TIM5_MMS0 ((uint8_t) (0x01 << 4))

  TIM5 Master mode selection [0] (in _TIM5_CR2)
- #define _TIM5_MMS1 ((uint8_t) (0x01 << 5))

  TIM5 Master mode selection [1] (in _TIM5_CR2)
- #define _TIM5_MMS2 ((uint8_t) (0x01 << 6))

  TIM5 Master mode selection [2] (in _TIM5_CR2)
- #define _TIM5_SMS ((uint8_t) (0x07 << 0))

  TIM5 Clock/trigger/slave mode selection [2:0] (in _TIM5_SMCR)
- #define _TIM5_SMS0 ((uint8_t) (0x01 << 0))

  TIM5 Clock/trigger/slave mode selection [0] (in _TIM5_SMCR)
- #define _TIM5_SMS1 ((uint8_t) (0x01 << 1))

  TIM5 Clock/trigger/slave mode selection [1] (in _TIM5_SMCR)
- #define _TIM5_SMS2 ((uint8_t) (0x01 << 2))

  TIM5 Clock/trigger/slave mode selection [2] (in _TIM5_SMCR)
- #define _TIM5_TS ((uint8_t) (0x07 << 4))

  TIM5 Trigger selection [2:0] (in _TIM5_SMCR)
- #define _TIM5_TS0 ((uint8_t) (0x01 << 4))

  TIM5 Trigger selection [0] (in _TIM5_SMCR)
- #define _TIM5_TS1 ((uint8_t) (0x01 << 5))

  TIM5 Trigger selection [1] (in _TIM5_SMCR)
- #define _TIM5_TS2 ((uint8_t) (0x01 << 6))

  TIM5 Trigger selection [2] (in _TIM5_SMCR)
- #define _TIM5_MSM ((uint8_t) (0x01 << 7))

  TIM5 Master/slave mode [0] (in _TIM5_SMCR)
- #define _TIM5_UIE ((uint8_t) (0x01 << 0))

  TIM5 Update interrupt enable [0] (in _TIM5_IER)
- #define _TIM5_CC1IE ((uint8_t) (0x01 << 1))

  TIM5 Capture/compare 1 interrupt enable [0] (in _TIM5_IER)
- #define _TIM5_CC2IE ((uint8_t) (0x01 << 2))

  TIM5 Capture/compare 2 interrupt enable [0] (in _TIM5_IER)
- #define _TIM5_CC3IE ((uint8_t) (0x01 << 3))

  TIM5 Capture/compare 3 interrupt enable [0] (in _TIM5_IER)
- #define _TIM5_TIE ((uint8_t) (0x01 << 6))

  TIM5 Trigger interrupt enable [0] (in _TIM5_IER)
- #define _TIM5_UIF ((uint8_t) (0x01 << 0))

  TIM5 Update interrupt flag [0] (in _TIM5_SR1)
- #define _TIM5_CC1IF ((uint8_t) (0x01 << 1))

  TIM5 Capture/compare 1 interrupt flag [0] (in _TIM5_SR1)
- #define _TIM5_CC2IF ((uint8_t) (0x01 << 2))

  TIM5 Capture/compare 2 interrupt flag [0] (in _TIM5_SR1)
- #define _TIM5_CC3IF ((uint8_t) (0x01 << 3))

  TIM5 Capture/compare 3 interrupt flag [0] (in _TIM5_SR1)
- #define _TIM5_TIF ((uint8_t) (0x01 << 6))

  TIM5 Trigger interrupt flag [0] (in _TIM5_SR1)

- #define _TIM5_CC1OF ((uint8_t) (0x01 << 1))
  TIM5 Capture/compare 1 overcapture flag [0] (in _TIM5_SR2)
- #define _TIM5_CC2OF ((uint8_t) (0x01 << 2))
  TIM5 Capture/compare 2 overcapture flag [0] (in _TIM5_SR2)
- #define _TIM5_CC3OF ((uint8_t) (0x01 << 3))
  TIM5 Capture/compare 3 overcapture flag [0] (in _TIM5_SR2)
- #define _TIM5_UG ((uint8_t) (0x01 << 0))
  TIM5 Update generation [0] (in _TIM5_EGR)
- #define _TIM5_CC1G ((uint8_t) (0x01 << 1))
  TIM5 Capture/compare 1 generation [0] (in _TIM5_EGR)
- #define _TIM5_CC2G ((uint8_t) (0x01 << 2))
  TIM5 Capture/compare 2 generation [0] (in _TIM5_EGR)
- #define _TIM5_CC3G ((uint8_t) (0x01 << 3))
  TIM5 Capture/compare 3 generation [0] (in _TIM5_EGR)
- #define _TIM5_TG ((uint8_t) (0x01 << 6))
  TIM5 Trigger generation [0] (in _TIM5_EGR)
- #define _TIM5_CC1S ((uint8_t) (0x03 << 0))
  TIM5 Compare 1 selection [1:0] (in _TIM5_CCMR1)
- #define _TIM5_CC1S0 ((uint8_t) (0x01 << 0))
  TIM5 Compare 1 selection [0] (in _TIM5_CCMR1)
- #define _TIM5_CC1S1 ((uint8_t) (0x01 << 1))
  TIM5 Compare 1 selection [1] (in _TIM5_CCMR1)
- #define _TIM5_OC1PE ((uint8_t) (0x01 << 3))
  TIM5 Output compare 1 preload enable [0] (in _TIM5_CCMR1)
- #define _TIM5_OC1M ((uint8_t) (0x07 << 4))
  TIM5 Output compare 1 mode [2:0] (in _TIM5_CCMR1)
- #define _TIM5_OC1M0 ((uint8_t) (0x01 << 4))
  TIM5 Output compare 1 mode [0] (in _TIM5_CCMR1)
- #define _TIM5_OC1M1 ((uint8_t) (0x01 << 5))
  TIM5 Output compare 1 mode [1] (in _TIM5_CCMR1)
- #define _TIM5_OC1M2 ((uint8_t) (0x01 << 6))
  TIM5 Output compare 1 mode [2] (in _TIM5_CCMR1)
- #define _TIM5_IC1PSC ((uint8_t) (0x03 << 2))
  TIM5 Input capture 1 prescaler [1:0] (in _TIM5_CCMR1)
- #define _TIM5_IC1PSC0 ((uint8_t) (0x01 << 2))
  TIM5 Input capture 1 prescaler [0] (in _TIM5_CCMR1)
- #define _TIM5_IC1PSC1 ((uint8_t) (0x01 << 3))
  TIM5 Input capture 1 prescaler [1] (in _TIM5_CCMR1)
- #define _TIM5_IC1F ((uint8_t) (0x0F << 4))
  TIM5 Output compare 1 mode [3:0] (in _TIM5_CCMR1)
- #define _TIM5_IC1F0 ((uint8_t) (0x01 << 4))
  TIM5 Input capture 1 filter [0] (in _TIM5_CCMR1)
- #define _TIM5_IC1F1 ((uint8_t) (0x01 << 5))
  TIM5 Input capture 1 filter [1] (in _TIM5_CCMR1)
- #define _TIM5_IC1F2 ((uint8_t) (0x01 << 6))
  TIM5 Input capture 1 filter [2] (in _TIM5_CCMR1)
- #define _TIM5_IC1F3 ((uint8_t) (0x01 << 7))
  TIM5 Input capture 1 filter [3] (in _TIM5_CCMR1)
- #define _TIM5_CC2S ((uint8_t) (0x03 << 0))
  TIM5 Compare 2 selection [1:0] (in _TIM5_CCMR2)
- #define _TIM5_CC2S0 ((uint8_t) (0x01 << 0))

TIM5 Compare 2 selection [0] (in _TIM5_CCMR2)

- #define _TIM5_CC2S1 ((uint8_t) (0x01 << 1))

  TIM5 Compare 2 selection [1] (in _TIM5_CCMR2)
- #define _TIM5_OC2PE ((uint8_t) (0x01 << 3))

  TIM5 Output compare 2 preload enable [0] (in _TIM5_CCMR2)
- #define _TIM5_OC2M ((uint8_t) (0x07 << 4))

  TIM5 Output compare 2 mode [2:0] (in _TIM5_CCMR2)
- #define _TIM5_OC2M0 ((uint8_t) (0x01 << 4))

  TIM5 Output compare 2 mode [0] (in _TIM5_CCMR2)
- #define _TIM5_OC2M1 ((uint8_t) (0x01 << 5))

  TIM5 Output compare 2 mode [1] (in _TIM5_CCMR2)
- #define _TIM5_OC2M2 ((uint8_t) (0x01 << 6))

  TIM5 Output compare 2 mode [2] (in _TIM5_CCMR2)
- #define _TIM5_IC2PSC ((uint8_t) (0x03 << 2))

  TIM5 Input capture 2 prescaler [1:0] (in _TIM5_CCMR2)
- #define _TIM5_IC2PSC0 ((uint8_t) (0x01 << 2))

  TIM5 Input capture 2 prescaler [0] (in _TIM5_CCMR2)
- #define _TIM5_IC2PSC1 ((uint8_t) (0x01 << 3))

  TIM5 Input capture 2 prescaler [1] (in _TIM5_CCMR2)
- #define _TIM5_IC2F ((uint8_t) (0x0F << 4))

  TIM5 Output compare 2 mode [3:0] (in _TIM5_CCMR2)
- #define _TIM5_IC2F0 ((uint8_t) (0x01 << 4))

  TIM5 Input capture 2 filter [0] (in _TIM5_CCMR2)
- #define _TIM5_IC2F1 ((uint8_t) (0x01 << 5))

  TIM5 Input capture 2 filter [1] (in _TIM5_CCMR2)
- #define _TIM5_IC2F2 ((uint8_t) (0x01 << 6))

  TIM5 Input capture 2 filter [2] (in _TIM5_CCMR2)
- #define _TIM5_IC2F3 ((uint8_t) (0x01 << 7))

  TIM5 Input capture 2 filter [3] (in _TIM5_CCMR2)
- #define _TIM5_CC3S ((uint8_t) (0x03 << 0))

  TIM5 Compare 3 selection [1:0] (in _TIM5_CCMR3)
- #define _TIM5_CC3S0 ((uint8_t) (0x01 << 0))

  TIM5 Compare 3 selection [0] (in _TIM5_CCMR3)
- #define _TIM5_CC3S1 ((uint8_t) (0x01 << 1))

  TIM5 Compare 3 selection [1] (in _TIM5_CCMR3)
- #define _TIM5_OC3PE ((uint8_t) (0x01 << 3))

  TIM5 Output compare 3 preload enable [0] (in _TIM5_CCMR3)
- #define _TIM5_OC3M ((uint8_t) (0x07 << 4))

  TIM5 Output compare 3 mode [2:0] (in _TIM5_CCMR3)
- #define _TIM5_OC3M0 ((uint8_t) (0x01 << 4))

  TIM5 Output compare 3 mode [0] (in _TIM5_CCMR3)
- #define _TIM5_OC3M1 ((uint8_t) (0x01 << 5))

  TIM5 Output compare 3 mode [1] (in _TIM5_CCMR3)
- #define _TIM5_OC3M2 ((uint8_t) (0x01 << 6))

  TIM5 Output compare 3 mode [2] (in _TIM5_CCMR3)
- #define _TIM5_IC3PSC ((uint8_t) (0x03 << 2))

  TIM5 Input capture 3 prescaler [1:0] (in _TIM5_CCMR3)
- #define _TIM5_IC3PSC0 ((uint8_t) (0x01 << 2))

  TIM5 Input capture 3 prescaler [0] (in _TIM5_CCMR3)
- #define _TIM5_IC3PSC1 ((uint8_t) (0x01 << 3))

  TIM5 Input capture 3 prescaler [1] (in _TIM5_CCMR3)

- #define _TIM5_IC3F ((uint8_t) (0x0F << 4))

  TIM5 Output compare 3 mode [3:0] (in _TIM5_CCMR3)
- #define _TIM5_IC3F0 ((uint8_t) (0x01 << 4))

  TIM5 Input capture 3 filter [0] (in _TIM5_CCMR3)
- #define _TIM5_IC3F1 ((uint8_t) (0x01 << 5))

  TIM5 Input capture 3 filter [1] (in _TIM5_CCMR3)
- #define _TIM5_IC3F2 ((uint8_t) (0x01 << 6))

  TIM5 Input capture 3 filter [2] (in _TIM5_CCMR3)
- #define _TIM5_IC3F3 ((uint8_t) (0x01 << 7))

  TIM5 Input capture 3 filter [3] (in _TIM5_CCMR3)
- #define _TIM5_CC1E ((uint8_t) (0x01 << 0))

  TIM5 Capture/compare 1 output enable [0] (in _TIM5_CCER1)
- #define _TIM5_CC1P ((uint8_t) (0x01 << 1))

  TIM5 Capture/compare 1 output polarity [0] (in _TIM5_CCER1)
- #define _TIM5_CC2E ((uint8_t) (0x01 << 4))

  TIM5 Capture/compare 2 output enable [0] (in _TIM5_CCER1)
- #define _TIM5_CC2P ((uint8_t) (0x01 << 5))

  TIM5 Capture/compare 2 output polarity [0] (in _TIM5_CCER1)
- #define _TIM5_CC3E ((uint8_t) (0x01 << 0))

  TIM5 Capture/compare 3 output enable [0] (in _TIM5_CCER2)
- #define _TIM5_CC3P ((uint8_t) (0x01 << 1))

  TIM5 Capture/compare 3 output polarity [0] (in _TIM5_CCER2)
- #define _TIM5_PSC ((uint8_t) (0x0F << 0))

  TIM5 clock prescaler [3:0] (in _TIM5_PSCR)
- #define _TIM5_PSC0 ((uint8_t) (0x01 << 0))

  TIM5 clock prescaler [0] (in _TIM5_PSCR)
- #define _TIM5_PSC1 ((uint8_t) (0x01 << 1))

  TIM5 clock prescaler [1] (in _TIM5_PSCR)
- #define _TIM5_PSC2 ((uint8_t) (0x01 << 2))

  TIM5 clock prescaler [2] (in _TIM5_PSCR)
- #define _TIM5_PSC3 ((uint8_t) (0x01 << 3))

  TIM5 clock prescaler [3] (in _TIM5_PSCR)
- #define _TIM6 _SFR(TIM6_t, TIM6_AddressBase)

  TIM6 struct/bit access.
- #define _TIM6_CR _SFR(uint8_t, TIM6_AddressBase+0x00)

  TIM6 control register.
- #define _TIM6_IER _SFR(uint8_t, TIM6_AddressBase+0x01)

  TIM6 interrupt enable register.
- #define _TIM6_SR _SFR(uint8_t, TIM6_AddressBase+0x02)

  TIM6 status register.
- #define _TIM6_EGR _SFR(uint8_t, TIM6_AddressBase+0x03)

  TIM6 event generation register.
- #define _TIM6_CNTR _SFR(uint8_t, TIM6_AddressBase+0x04)

  TIM6 counter register.
- #define _TIM6_PSCR _SFR(uint8_t, TIM6_AddressBase+0x05)

  TIM6 clock prescaler register.
- #define _TIM6_ARR _SFR(uint8_t, TIM6_AddressBase+0x06)

  TIM6 auto-reload register.
- #define _TIM6_CR_RESET_VALUE ((uint8_t) 0x00)

  TIM6 control register reset value.
- #define _TIM6_IER_RESET_VALUE ((uint8_t) 0x00)

TIM6 interrupt enable register reset value.

- #define _TIM6_SR_RESET_VALUE ((uint8_t) 0x00)

  TIM6 status register reset value.
- #define _TIM6_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM6 event generation register reset value.
- #define _TIM6_CNTR_RESET_VALUE ((uint8_t) 0x00)

  TIM6 counter register reset value.
- #define _TIM6_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM6 clock prescaler register reset value.
- #define _TIM6_ARR_RESET_VALUE ((uint8_t) 0xFF)

  TIM6 auto-reload register reset value.
- #define _TIM6_CEN ((uint8_t) (0x01 << 0))

  TIM6 Counter enable [0] (in _TIM6_CR1)
- #define _TIM6_UDIS ((uint8_t) (0x01 << 1))

  TIM6 Update disable [0] (in _TIM6_CR1)
- #define _TIM6_URS ((uint8_t) (0x01 << 2))

  TIM6 Update request source [0] (in _TIM6_CR1)
- #define _TIM6_OPM ((uint8_t) (0x01 << 3))

  TIM6 One-pulse mode [0] (in _TIM6_CR1)
- #define _TIM6_ARPE ((uint8_t) (0x01 << 7))

  TIM6 Auto-reload preload enable [0] (in _TIM6_CR1)
- #define _TIM6_MMS ((uint8_t) (0x07 << 4))

  TIM6 Master mode selection [2:0] (in _TIM6_CR2)
- #define _TIM6_MMS0 ((uint8_t) (0x01 << 4))

  TIM6 Master mode selection [0] (in _TIM6_CR2)
- #define _TIM6_MMS1 ((uint8_t) (0x01 << 5))

  TIM6 Master mode selection [1] (in _TIM6_CR2)
- #define _TIM6_MMS2 ((uint8_t) (0x01 << 6))

  TIM6 Master mode selection [2] (in _TIM6_CR2)
- #define _TIM6_SMS ((uint8_t) (0x07 << 0))

  TIM6 Clock/trigger/slave mode selection [2:0] (in _TIM6_SMCR)
- #define _TIM6_SMS0 ((uint8_t) (0x01 << 0))

  TIM6 Clock/trigger/slave mode selection [0] (in _TIM6_SMCR)
- #define _TIM6_SMS1 ((uint8_t) (0x01 << 1))

  TIM6 Clock/trigger/slave mode selection [1] (in _TIM6_SMCR)
- #define _TIM6_SMS2 ((uint8_t) (0x01 << 2))

  TIM6 Clock/trigger/slave mode selection [2] (in _TIM6_SMCR)
- #define _TIM6_TS ((uint8_t) (0x07 << 4))

  TIM6 Trigger selection [2:0] (in _TIM6_SMCR)
- #define _TIM6_TS0 ((uint8_t) (0x01 << 4))

  TIM6 Trigger selection [0] (in _TIM6_SMCR)
- #define _TIM6_TS1 ((uint8_t) (0x01 << 5))

  TIM6 Trigger selection [1] (in _TIM6_SMCR)
- #define _TIM6_TS2 ((uint8_t) (0x01 << 6))

  TIM6 Trigger selection [2] (in _TIM6_SMCR)
- #define _TIM6_UIE ((uint8_t) (0x01 << 0))

  TIM6 Update interrupt enable [0] (in _TIM6_IER)
- #define _TIM6_UIF ((uint8_t) (0x01 << 0))

  TIM6 Update interrupt flag [0] (in _TIM6_SR)
- #define _TIM6_UG ((uint8_t) (0x01 << 0))

  TIM6 Update generation [0] (in _TIM6_EGR)

- #define _TIM6_PSC ((uint8_t) (0x07 << 0))
  TIM6 clock prescaler [2:0] (in _TIM6_PSCR)
- #define _TIM6_PSC0 ((uint8_t) (0x01 << 0))
  TIM6 clock prescaler [0] (in _TIM6_PSCR)
- #define _TIM6_PSC1 ((uint8_t) (0x01 << 1))
  TIM6 clock prescaler [1] (in _TIM6_PSCR)
- #define _TIM6_PSC2 ((uint8_t) (0x01 << 2))
  TIM6 clock prescaler [2] (in _TIM6_PSCR)
- #define _ADC1 _SFR(ADC1_t, ADC1_AddressBase)
  ADC1 struct/bit access.
- #define _ADC1_DB0RH _SFR(uint8_t, ADC1_AddressBase+0x00)
  ADC1 10-bit Data Buffer Register 0.
- #define _ADC1_DB0RL _SFR(uint8_t, ADC1_AddressBase+0x01)
  ADC1 10-bit Data Buffer Register 0.
- #define _ADC1_DB1RH _SFR(uint8_t, ADC1_AddressBase+0x02)
  ADC1 10-bit Data Buffer Register 1.
- #define _ADC1_DB1RL _SFR(uint8_t, ADC1_AddressBase+0x03)
  ADC1 10-bit Data Buffer Register 1.
- #define _ADC1_DB2RH _SFR(uint8_t, ADC1_AddressBase+0x04)
  ADC1 10-bit Data Buffer Register 2.
- #define _ADC1_DB2RL _SFR(uint8_t, ADC1_AddressBase+0x05)
  ADC1 10-bit Data Buffer Register 2.
- #define _ADC1_DB3RH _SFR(uint8_t, ADC1_AddressBase+0x06)
  ADC1 10-bit Data Buffer Register 3.
- #define _ADC1_DB3RL _SFR(uint8_t, ADC1_AddressBase+0x07)
  ADC1 10-bit Data Buffer Register 3.
- #define _ADC1_DB4RH _SFR(uint8_t, ADC1_AddressBase+0x08)
  ADC1 10-bit Data Buffer Register 4.
- #define _ADC1_DB4RL _SFR(uint8_t, ADC1_AddressBase+0x09)
  ADC1 10-bit Data Buffer Register 4.
- #define _ADC1_DB5RH _SFR(uint8_t, ADC1_AddressBase+0x0A)
  ADC1 10-bit Data Buffer Register 5.
- #define _ADC1_DB5RL _SFR(uint8_t, ADC1_AddressBase+0x0B)
  ADC1 10-bit Data Buffer Register 5.
- #define _ADC1_DB6RH _SFR(uint8_t, ADC1_AddressBase+0x0C)
  ADC1 10-bit Data Buffer Register 6.
- #define _ADC1_DB6RL _SFR(uint8_t, ADC1_AddressBase+0x0D)
  ADC1 10-bit Data Buffer Register 6.
- #define _ADC1_DB7RH _SFR(uint8_t, ADC1_AddressBase+0x0E)
  ADC1 10-bit Data Buffer Register 7.
- #define _ADC1_DB7RL _SFR(uint8_t, ADC1_AddressBase+0x0F)
  ADC1 10-bit Data Buffer Register 7.
- #define _ADC1_DB8RH _SFR(uint8_t, ADC1_AddressBase+0x10)
  ADC1 10-bit Data Buffer Register 8.
- #define _ADC1_DB8RL _SFR(uint8_t, ADC1_AddressBase+0x11)
  ADC1 10-bit Data Buffer Register 8.
- #define _ADC1_DB9RH _SFR(uint8_t, ADC1_AddressBase+0x12)
  ADC1 10-bit Data Buffer Register 9.
- #define _ADC1_DB9RL _SFR(uint8_t, ADC1_AddressBase+0x13)
  ADC1 10-bit Data Buffer Register 9.
- #define _ADC1_CSR _SFR(uint8_t, ADC1_AddressBase+0x20)

ADC1 control/status register.

- #define _ADC1_CR1 _SFR(uint8_t, ADC1_AddressBase+0x21)

  ADC1 Configuration Register 1.
- #define _ADC1_CR2 _SFR(uint8_t, ADC1_AddressBase+0x22)

  ADC1 Configuration Register 2.
- #define _ADC1_CR3 _SFR(uint8_t, ADC1_AddressBase+0x23)

  ADC1 Configuration Register 3.
- #define _ADC1_DRH _SFR(uint8_t, ADC1_AddressBase+0x24)

  ADC1 (unbuffered) 10-bit measurement result.
- #define _ADC1_DRL _SFR(uint8_t, ADC1_AddressBase+0x25)

  ADC1 (unbuffered) 10-bit measurement result.
- #define _ADC1_TDRH _SFR(uint8_t, ADC1_AddressBase+0x26)

  ADC1 Schmitt trigger disable register.
- #define _ADC1_TDRL _SFR(uint8_t, ADC1_AddressBase+0x27)

  ADC1 Schmitt trigger disable register.
- #define _ADC1_HTRH _SFR(uint8_t, ADC1_AddressBase+0x28)

  ADC1 watchdog high threshold register.
- #define _ADC1_HTRL _SFR(uint8_t, ADC1_AddressBase+0x29)

  ADC1 watchdog high threshold register.
- #define _ADC1_LTRH _SFR(uint8_t, ADC1_AddressBase+0x2A)

  ADC1 watchdog low threshold register.
- #define _ADC1_LTRL _SFR(uint8_t, ADC1_AddressBase+0x2B)

  ADC1 watchdog low threshold register.
- #define _ADC1_AWSRH _SFR(uint8_t, ADC1_AddressBase+0x2C)

  ADC1 watchdog status register.
- #define _ADC1_AWSRL _SFR(uint8_t, ADC1_AddressBase+0x2D)

  ADC1 watchdog status register.
- #define _ADC1_AWCRH _SFR(uint8_t, ADC1_AddressBase+0x2E)

  ADC1 watchdog control register.
- #define _ADC1_AWCRL _SFR(uint8_t, ADC1_AddressBase+0x2F)

  ADC1 watchdog control register.
- #define _ADC1_CSR_RESET_VALUE ((uint8_t) 0x00)

  ADC1 control/status register reset value.
- #define _ADC1_CR1_RESET_VALUE ((uint8_t) 0x00)

  ADC1 Configuration Register 1 reset value.
- #define _ADC1_CR2_RESET_VALUE ((uint8_t) 0x00)

  ADC1 Configuration Register 2 reset value.
- #define _ADC1_CR3_RESET_VALUE ((uint8_t) 0x00)

  ADC1 Configuration Register 3 reset value.
- #define _ADC1_TDRH_RESET_VALUE ((uint8_t) 0x00)

  ADC1 Schmitt trigger disable register reset value.
- #define _ADC1_TDRL_RESET_VALUE ((uint8_t) 0x00)

  ADC1 Schmitt trigger disable register reset value.
- #define _ADC1_HTRH_RESET_VALUE ((uint8_t) 0xFF)

  ADC1 watchdog high threshold register reset value.
- #define _ADC1_HTRL_RESET_VALUE ((uint8_t) 0x03)

  ADC1 watchdog high threshold register reset value.
- #define _ADC1_LTRH_RESET_VALUE ((uint8_t) 0x00)

  ADC1 watchdog low threshold register reset value.
- #define _ADC1_LTRL_RESET_VALUE ((uint8_t) 0x00)

  ADC1 watchdog low threshold register reset value.

- #define _ADC1_AWCRH_RESET_VALUE ((uint8_t) 0x00)

  ADC1 watchdog control register reset value.
- #define _ADC1_AWCRL_RESET_VALUE ((uint8_t) 0x00)

  ADC1 watchdog control register reset value.
- #define _ADC1_CH ((uint8_t) (0x0F << 0))

  ADC1 Channel selection bits [3:0] (in _ADC1_CSR)
- #define _ADC1_CH0 ((uint8_t) (0x01 << 0))

  ADC1 Channel selection bits [0] (in _ADC1_CSR)
- #define _ADC1_CH1 ((uint8_t) (0x01 << 1))

  ADC1 Channel selection bits [1] (in _ADC1_CSR)
- #define _ADC1_CH2 ((uint8_t) (0x01 << 2))

  ADC1 Channel selection bits [2] (in _ADC1_CSR)
- #define _ADC1_CH3 ((uint8_t) (0x01 << 3))

  ADC1 Channel selection bits [3] (in _ADC1_CSR)
- #define _ADC1_AWDIE ((uint8_t) (0x01 << 4))

  ADC1 Analog watchdog interrupt enable [0] (in _ADC1_CSR)
- #define _ADC1_EOCIE ((uint8_t) (0x01 << 5))

  ADC1 Interrupt enable for EOC [0] (in _ADC1_CSR)
- #define _ADC1_AWD ((uint8_t) (0x01 << 6))

  ADC1 Analog Watchdog flag [0] (in _ADC1_CSR)
- #define _ADC1_EOC ((uint8_t) (0x01 << 7))

  ADC1 End of conversion [0] (in _ADC1_CSR)
- #define _ADC1_ADON ((uint8_t) (0x01 << 0))

  ADC1 Conversion on/off [0] (in _ADC1_CR1)
- #define _ADC1_CONT ((uint8_t) (0x01 << 1))

  ADC1 Continuous conversion [0] (in _ADC1_CR1)
- #define _ADC1_SPSEL ((uint8_t) (0x07 << 4)

  ADC1 clock prescaler selection [2:0] (in _ADC1_CR1)
- #define _ADC1_SPSEL0 ((uint8_t) (0x01 << 4)

  ADC1 clock prescaler selection [0] (in _ADC1_CR1)
- #define _ADC1_SPSEL1 ((uint8_t) (0x01 << 5)

  ADC1 clock prescaler selection [1] (in _ADC1_CR1)
- #define _ADC1_SPSEL2 ((uint8_t) (0x01 << 6)

  ADC1 clock prescaler selection [2] (in _ADC1_CR1)
- #define _ADC1_SCAN ((uint8_t) (0x01 << 1))

  ADC1 Scan mode enable [0] (in _ADC1_CR2)
- #define _ADC1_ALIGN ((uint8_t) (0x01 << 3))

  ADC1 Data alignment [0] (in _ADC1_CR2)
- #define _ADC1_EXTSEL ((uint8_t) (0x03 << 4)

  ADC1 External event selection [1:0] (in _ADC1_CR2)
- #define _ADC1_EXTSEL0 ((uint8_t) (0x01 << 4)

  ADC1 External event selection [0] (in _ADC1_CR2)
- #define _ADC1_EXTSEL1 ((uint8_t) (0x01 << 5)

  ADC1 External event selection [1] (in _ADC1_CR2)
- #define _ADC1_EXTTRIG ((uint8_t) (0x01 << 6))

  ADC1 External trigger enable [0] (in _ADC1_CR2)
- #define _ADC1_OVR ((uint8_t) (0x01 << 6))

  ADC1 Overrun flag [0] (in _ADC1_CR3)
- #define _ADC1_DBUF ((uint8_t) (0x01 << 7))

  ADC1 Data buffer enable [0] (in _ADC1_CR3)
- #define _ADC2 _SFR(ADC2_t, ADC2_AddressBase)

ADC2 struct/bit access.

- #define _ADC2_CSR _SFR(uint8_t, ADC2_AddressBase+0x00)

  ADC2 control/status register.
- #define _ADC2_CR1 _SFR(uint8_t, ADC2_AddressBase+0x01)

  ADC2 Configuration Register 1.
- #define _ADC2_CR2 _SFR(uint8_t, ADC2_AddressBase+0x02)

  ADC2 Configuration Register 2.
- #define _ADC2_DRH _SFR(uint8_t, ADC2_AddressBase+0x04)

  ADC2 (unbuffered) 10-bit measurement result.
- #define _ADC2_DRL _SFR(uint8_t, ADC2_AddressBase+0x05)

  ADC2 (unbuffered) 10-bit measurement result.
- #define _ADC2_TDRH _SFR(uint8_t, ADC2_AddressBase+0x06)

  ADC2 Schmitt trigger disable register.
- #define _ADC2_TDRL _SFR(uint8_t, ADC2_AddressBase+0x07)

  ADC2 Schmitt trigger disable register.
- #define _ADC2_CSR_RESET_VALUE ((uint8_t) 0x00)

  ADC2 control/status register reset value.
- #define _ADC2_CR1_RESET_VALUE ((uint8_t) 0x00)

  ADC2 Configuration Register 1 reset value.
- #define _ADC2_CR2_RESET_VALUE ((uint8_t) 0x00)

  ADC2 Configuration Register 2 reset value.
- #define _ADC2_TDRL_RESET_VALUE ((uint8_t) 0x00)

  ADC2 Schmitt trigger disable register reset value.
- #define _ADC2_TDRH_RESET_VALUE ((uint8_t) 0x00)

  ADC2 Schmitt trigger disable register reset value.
- #define _ADC2_CH ((uint8_t) (0x0F << 0))

  ADC2 Channel selection bits [3:0] (in _ADC2_CSR)
- #define _ADC2_CH0 ((uint8_t) (0x01 << 0))

  ADC2 Channel selection bits [0] (in _ADC2_CSR)
- #define _ADC2_CH1 ((uint8_t) (0x01 << 1))

  ADC2 Channel selection bits [1] (in _ADC2_CSR)
- #define _ADC2_CH2 ((uint8_t) (0x01 << 2))

  ADC2 Channel selection bits [2] (in _ADC2_CSR)
- #define _ADC2_CH3 ((uint8_t) (0x01 << 3))

  ADC2 Channel selection bits [3] (in _ADC2_CSR)
- #define _ADC2_EOCIE ((uint8_t) (0x01 << 5))

  ADC2 Interrupt enable for EOC [0] (in _ADC2_CSR)
- #define _ADC2_EOC ((uint8_t) (0x01 << 7))

  ADC2 End of conversion [0] (in _ADC2_CSR)
- #define _ADC2_ADON ((uint8_t) (0x01 << 0))

  ADC2 Conversion on/off [0] (in _ADC2_CR1)
- #define _ADC2_CONT ((uint8_t) (0x01 << 1))

  ADC2 Continuous conversion [0] (in _ADC2_CR1)
- #define _ADC2_SPSEL ((uint8_t) (0x07 << 4)

  ADC2 clock prescaler selection [2:0] (in _ADC2_CR1)
- #define _ADC2_SPSEL0 ((uint8_t) (0x01 << 4)

  ADC2 clock prescaler selection [0] (in _ADC2_CR1)
- #define _ADC2_SPSEL1 ((uint8_t) (0x01 << 5)

  ADC2 clock prescaler selection [1] (in _ADC2_CR1)
- #define _ADC2_SPSEL2 ((uint8_t) (0x01 << 6)

  ADC2 clock prescaler selection [2] (in _ADC2_CR1)

- #define _ADC2_ALIGN ((uint8_t) (0x01 << 3))

  ADC2 Data alignment [0] (in _ADC2_CR2)
- #define _ADC2_EXTSEL ((uint8_t) (0x03 << 4)

  ADC2 External event selection [1:0] (in _ADC2_CR2)
- #define _ADC2_EXTSEL0 ((uint8_t) (0x01 << 4)

  ADC2 External event selection [0] (in _ADC2_CR2)
- #define _ADC2_EXTSEL1 ((uint8_t) (0x01 << 5)

  ADC2 External event selection [1] (in _ADC2_CR2)
- #define _ADC2_EXTTRIG ((uint8_t) (0x01 << 6))

  ADC2 External trigger enable [0] (in _ADC2_CR2)
- #define _CAN _SFR(CAN_t, CAN_AddressBase)

  CAN struct/bit access.
- #define _CAN_MCR _SFR(uint8_t, CAN_AddressBase+0x00)

  CAN master control register.
- #define _CAN_MSR _SFR(uint8_t, CAN_AddressBase+0x01)

  CAN master status register.
- #define _CAN_TSR _SFR(uint8_t, CAN_AddressBase+0x02)

  CAN transmit status register.
- #define _CAN_TPR _SFR(uint8_t, CAN_AddressBase+0x03)

  CAN transmit priority register.
- #define _CAN_RFR _SFR(uint8_t, CAN_AddressBase+0x04)

  CAN receive FIFO register.
- #define _CAN_IER _SFR(uint8_t, CAN_AddressBase+0x05)

  CAN interrupt enable register.
- #define _CAN_DGR _SFR(uint8_t, CAN_AddressBase+0x06)

  CAN diagnosis register.
- #define _CAN_PSR _SFR(uint8_t, CAN_AddressBase+0x07)

  CAN page selection for below paged registers.
- #define _CAN_MCSR _SFR(uint8_t, CAN_AddressBase+0x08+0x00)

  CAN message control/status register (page 0,1,5)
- #define _CAN_MDLCR _SFR(uint8_t, CAN_AddressBase+0x08+0x01)

  CAN mailbox data length control register (page 0,1,5,7)
- #define _CAN_MIDR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x02)

  CAN mailbox identifier register 1 (page 0,1,5,7)
- #define _CAN_MIDR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x03)

  CAN mailbox identifier register 2 (page 0,1,5,7)
- #define _CAN_MIDR3 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)

  CAN mailbox identifier register 3 (page 0,1,5,7)
- #define _CAN_MIDR4 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)

  CAN mailbox identifier register 4 (page 0,1,5,7)
- #define _CAN_MDAR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x06)

  *CAN mailbox data register 1 (page 0,1,5,7) */.*
- #define _CAN_MDAR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x07)

  *CAN mailbox data register 2 (page 0,1,5,7) */.*
- #define _CAN_MDAR3 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)

  *CAN mailbox data register 3 (page 0,1,5,7) */.*
- #define _CAN_MDAR4 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)

  *CAN mailbox data register 4 (page 0,1,5,7) */.*
- #define _CAN_MDAR5 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)

  *CAN mailbox data register 5 (page 0,1,5,7) */.*
- #define _CAN_MDAR6 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)

*CAN mailbox data register 6 (page 0,1,5,7) */.*

- #define _CAN_MDAR7 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)

  *CAN mailbox data register 7 (page 0,1,5,7) */.*
- #define _CAN_MDAR8 _SFR(uint8_t, CAN_AddressBase+0x08+0x0D)

  *CAN mailbox data register 8 (page 0,1,5,7) */.*
- #define _CAN_MTSRL _SFR(uint8_t, CAN_AddressBase+0x08+0x0E)

  *CAN mailbox time stamp register low byte (page 0,1,5,7) */.*
- #define _CAN_MTSRH _SFR(uint8_t, CAN_AddressBase+0x08+0x0F)

  *CAN mailbox time stamp register high byte (page 0,1,5,7) */.*
- #define _CAN_F0R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x00)

  CAN acceptance filter 0/1 (page 2)
- #define _CAN_F0R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x01)

  CAN acceptance filter 0/2 (page 2)
- #define _CAN_F0R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x02)

  CAN acceptance filter 0/3 (page 2)
- #define _CAN_F0R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x03)

  CAN acceptance filter 0/4 (page 2)
- #define _CAN_F0R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)

  CAN acceptance filter 0/5 (page 2)
- #define _CAN_F0R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)

  CAN acceptance filter 0/6 (page 2)
- #define _CAN_F0R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x06)

  CAN acceptance filter 0/7 (page 2)
- #define _CAN_F0R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x07)

  CAN acceptance filter 0/8 (page 2)
- #define _CAN_F1R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)

  CAN acceptance filter 1/1 (page 2)
- #define _CAN_F1R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)

  CAN acceptance filter 1/2 (page 2)
- #define _CAN_F1R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)

  CAN acceptance filter 1/3 (page 2)
- #define _CAN_F1R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)

  CAN acceptance filter 1/4 (page 2)
- #define _CAN_F1R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)

  CAN acceptance filter 1/5 (page 2)
- #define _CAN_F1R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x0D)

  CAN acceptance filter 1/6 (page 2)
- #define _CAN_F1R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x0E)

  CAN acceptance filter 1/7 (page 2)
- #define _CAN_F1R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x0F)

  CAN acceptance filter 1/8 (page 2)
- #define _CAN_F2R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x00)

  CAN acceptance filter 2/1 (page 3)
- #define _CAN_F2R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x01)

  CAN acceptance filter 2/2 (page 3)
- #define _CAN_F2R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x02)

  CAN acceptance filter 2/3 (page 3)
- #define _CAN_F2R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x03)

  CAN acceptance filter 2/4 (page 3)
- #define _CAN_F2R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)

  CAN acceptance filter 2/5 (page 3)

- #define _CAN_F2R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)
  CAN acceptance filter 2/6 (page 3)
- #define _CAN_F2R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x06)
  CAN acceptance filter 2/7 (page 3)
- #define _CAN_F2R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x07)
  CAN acceptance filter 2/8 (page 3)
- #define _CAN_F3R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)
  CAN acceptance filter 3/1 (page 3)
- #define _CAN_F3R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)
  CAN acceptance filter 3/2 (page 3)
- #define _CAN_F3R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)
  CAN acceptance filter 3/3 (page 3)
- #define _CAN_F3R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)
  CAN acceptance filter 3/4 (page 3)
- #define _CAN_F3R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)
  CAN acceptance filter 3/5 (page 3)
- #define _CAN_F3R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x0D)
  CAN acceptance filter 3/6 (page 3)
- #define _CAN_F3R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x0E)
  CAN acceptance filter 3/7 (page 3)
- #define _CAN_F3R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x0F)
  CAN acceptance filter 3/8 (page 3)
- #define _CAN_F4R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x00)
  CAN acceptance filter 4/1 (page 4)
- #define _CAN_F4R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x01)
  CAN acceptance filter 4/2 (page 4)
- #define _CAN_F4R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x02)
  CAN acceptance filter 4/3 (page 4)
- #define _CAN_F4R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x03)
  CAN acceptance filter 4/4 (page 4)
- #define _CAN_F4R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)
  CAN acceptance filter 4/5 (page 4)
- #define _CAN_F4R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)
  CAN acceptance filter 4/6 (page 4)
- #define _CAN_F4R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x06)
  CAN acceptance filter 4/7 (page 4)
- #define _CAN_F4R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x07)
  CAN acceptance filter 4/8 (page 4)
- #define _CAN_F5R1 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)
  CAN acceptance filter 5/1 (page 4)
- #define _CAN_F5R2 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)
  CAN acceptance filter 5/2 (page 4)
- #define _CAN_F5R3 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)
  CAN acceptance filter 5/3 (page 4)
- #define _CAN_F5R4 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)
  CAN acceptance filter 5/4 (page 4)
- #define _CAN_F5R5 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)
  CAN acceptance filter 5/5 (page 4)
- #define _CAN_F5R6 _SFR(uint8_t, CAN_AddressBase+0x08+0x0D)
  CAN acceptance filter 5/6 (page 4)
- #define _CAN_F5R7 _SFR(uint8_t, CAN_AddressBase+0x08+0x0E)

CAN acceptance filter 5/7 (page 4)

- #define _CAN_F5R8 _SFR(uint8_t, CAN_AddressBase+0x08+0x0F)

  CAN acceptance filter 5/8 (page 4)
- #define _CAN_ESR _SFR(uint8_t, CAN_AddressBase+0x08+0x00)

  CAN error status register (page 6)
- #define _CAN_EIER _SFR(uint8_t, CAN_AddressBase+0x08+0x01)

  CAN error interrupt enable register (page 6)
- #define _CAN_TECR _SFR(uint8_t, CAN_AddressBase+0x08+0x02)

  CAN transmit error counter register (page 6)
- #define _CAN_RECR _SFR(uint8_t, CAN_AddressBase+0x08+0x03)

  CAN receive error counter register (page 6)
- #define _CAN_BTR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x04)

  CAN bit timing register 1 (page 6)
- #define _CAN_BTR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x05)

  CAN bit timing register 2 (page 6)
- #define _CAN_FMR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x08)

  CAN filter mode register 1 (page 6)
- #define _CAN_FMR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x09)

  CAN filter mode register 2 (page 6)
- #define _CAN_FCR1 _SFR(uint8_t, CAN_AddressBase+0x08+0x0A)

  CAN filter configuration register 1 (page 6)
- #define _CAN_FCR2 _SFR(uint8_t, CAN_AddressBase+0x08+0x0B)

  CAN filter configuration register 2 (page 6)
- #define _CAN_FCR3 _SFR(uint8_t, CAN_AddressBase+0x08+0x0C)

  CAN filter configuration register 3 (page 6)
- #define _CAN_MFMIR _SFR(uint8_t, CAN_AddressBase+0x08+0x00)

  CAN mailbox filter match index register (page 7)
- #define _CAN_MCR_RESET_VALUE ((uint8_t) 0x02)

  CAN master control register reset value.
- #define _CAN_MSR_RESET_VALUE ((uint8_t) 0x02)

  CAN master status register reset value.
- #define _CAN_TSR_RESET_VALUE ((uint8_t) 0x00)

  CAN transmit status register reset value.
- #define _CAN_TPR_RESET_VALUE ((uint8_t) 0x0C)

  CAN transmit priority register reset value.
- #define _CAN_RFR_RESET_VALUE ((uint8_t) 0x00)

  CAN receive FIFO register reset value.
- #define _CAN_IER_RESET_VALUE ((uint8_t) 0x00)

  CAN interrupt enable register reset value.
- #define _CAN_DGR_RESET_VALUE ((uint8_t) 0x0C)

  CAN diagnosis register reset value.
- #define _CAN_PSR_RESET_VALUE ((uint8_t) 0x00)

  CAN page selection reset value.
- #define _CAN_MCSR_RESET_VALUE ((uint8_t) 0x00)

  CAN message control/status register (page 0,1,5) reset value.
- #define _CAN_MDLCR_RESET_VALUE ((uint8_t) 0x00)

  CAN mailbox data length control register (page 0,1,5,7) reset value.
- #define _CAN_ESR_RESET_VALUE ((uint8_t) 0x00)

  CAN error status register (page 6) reset value.
- #define _CAN_EIER_RESET_VALUE ((uint8_t) 0x00)

  CAN error interrupt enable register (page 6) reset value.

- #define _CAN_TECR_RESET_VALUE ((uint8_t) 0x00)
  CAN transmit error counter register (page 6) reset value.
- #define _CAN_RECR_RESET_VALUE ((uint8_t) 0x00)
  CAN receive error counter register (page 6) reset value.
- #define _CAN_BTR1_RESET_VALUE ((uint8_t) 0x40)
  CAN bit timing register 1 (page 6) reset value.
- #define _CAN_BTR2_RESET_VALUE ((uint8_t) 0x23)
  CAN bit timing register 2 (page 6) reset value.
- #define _CAN_FMR1_RESET_VALUE ((uint8_t) 0x00)
  CAN filter mode register 1 (page 6) reset value.
- #define _CAN_FMR2_RESET_VALUE ((uint8_t) 0x00)
  CAN filter mode register 2 (page 6) reset value.
- #define _CAN_FCR_RESET_VALUE ((uint8_t) 0x00)
  CAN filter configuration register reset value.
- #define _CAN_MFMIR_RESET_VALUE ((uint8_t) 0x00)
  CAN mailbox filter match index register reset value.
- #define _CAN_INRQ ((uint8_t) (0x01 << 0))
  CAN Channel Initialization Request [0] (in _CAN_MCR)
- #define _CAN_SLEEP ((uint8_t) (0x01 << 1))
  CAN Channel Sleep Mode Request [0] (in _CAN_MCR)
- #define _CAN_TXFP ((uint8_t) (0x01 << 2))
  CAN Channel Transmit FIFO Priority [0] (in _CAN_MCR)
- #define _CAN_RFLM ((uint8_t) (0x01 << 3))
  CAN Channel Receive FIFO Locked Mode [0] (in _CAN_MCR)
- #define _CAN_NART ((uint8_t) (0x01 << 4))
  CAN Channel No Automatic Retransmission [0] (in _CAN_MCR)
- #define _CAN_AWUM ((uint8_t) (0x01 << 5))
  CAN Channel Automatic Wakeup Mode [0] (in _CAN_MCR)
- #define _CAN_ABOM ((uint8_t) (0x01 << 6))
  CAN Channel Automatic Bus-Off Management [0] (in _CAN_MCR)
- #define _CAN_TTCM ((uint8_t) (0x01 << 7))
  CAN Channel Time Triggered Communication Mode [0] (in _CAN_MCR)
- #define _CAN_INAK ((uint8_t) (0x01 << 0))
  CAN Initialization Acknowledge [0] (in _CAN_MSR)
- #define _CAN_SLAK ((uint8_t) (0x01 << 1))
  CAN Sleep Acknowledge [0] (in _CAN_MSR)
- #define _CAN_ERRI ((uint8_t) (0x01 << 2))
  CAN Error Interrupt [0] (in _CAN_MSR)
- #define _CAN_WKUI ((uint8_t) (0x01 << 3))
  CAN Wakeup Interrupt [0] (in _CAN_MSR)
- #define _CAN_TX ((uint8_t) (0x01 << 4))
  CAN Transmit [0] (in _CAN_MSR)
- #define _CAN_RX ((uint8_t) (0x01 << 5))
  CAN Receive [0] (in _CAN_MSR)
- #define _CAN_RQCP0 ((uint8_t) (0x01 << 0))
  CAN Request Completed for Mailbox 0 [0] (in _CAN_TSR)
- #define _CAN_RQCP1 ((uint8_t) (0x01 << 1))
  CAN Request Completed for Mailbox 1 [0] (in _CAN_TSR)
- #define _CAN_RQCP2 ((uint8_t) (0x01 << 2))
  CAN Request Completed for Mailbox 2 [0] (in _CAN_TSR)
- #define _CAN_TXOK0 ((uint8_t) (0x01 << 4))

CAN Transmission ok for Mailbox 0 [0] (in _CAN_TSR)

- #define _CAN_TXOK1 ((uint8_t) (0x01 << 5))

  CAN Transmission ok for Mailbox 1 [0] (in _CAN_TSR)
- #define _CAN_TXOK2 ((uint8_t) (0x01 << 6))

  CAN Transmission ok for Mailbox 2 [0] (in _CAN_TSR)
- #define _CAN_CODE ((uint8_t) (0x03 << 0))

  CAN Mailbox Code [1:0] (in _CAN_TPR)
- #define _CAN_CODE0 ((uint8_t) (0x01 << 0))

  CAN Mailbox Code [0] (in _CAN_TPR)
- #define _CAN_CODE1 ((uint8_t) (0x01 << 1))

  CAN Mailbox Code [1] (in _CAN_TPR)
- #define _CAN_TME0 ((uint8_t) (0x01 << 2))

  CAN Transmit Mailbox 0 Empty [0] (in _CAN_TPR)
- #define _CAN_TME1 ((uint8_t) (0x01 << 3))

  CAN Transmit Mailbox 1 Empty [0] (in _CAN_TPR)
- #define _CAN_TME2 ((uint8_t) (0x01 << 4))

  CAN Transmit Mailbox 2 Empty [0] (in _CAN_TPR)
- #define _CAN_LOW0 ((uint8_t) (0x01 << 5))

  CAN Lowest Priority Flag for Mailbox 0 [0] (in _CAN_TPR)
- #define _CAN_LOW1 ((uint8_t) (0x01 << 6))

  CAN Lowest Priority Flag for Mailbox 1 [0] (in _CAN_TPR)
- #define _CAN_LOW2 ((uint8_t) (0x01 << 7))

  CAN Lowest Priority Flag for Mailbox 2 [0] (in _CAN_TPR)
- #define _CAN_FMP ((uint8_t) (0x03 << 0))

  CAN FIFO Message Pending [1:0] (in _CAN_RFR)
- #define _CAN_FMP0 ((uint8_t) (0x01 << 0))

  CAN FIFO Message Pending [0] (in _CAN_RFR)
- #define _CAN_FMP1 ((uint8_t) (0x01 << 1))

  CAN FIFO Message Pending [1] (in _CAN_RFR)
- #define _CAN_FULL ((uint8_t) (0x01 << 3))

  CAN FIFO Full [0] (in _CAN_RFR)
- #define _CAN_FOVR ((uint8_t) (0x01 << 4))

  CAN FIFO Overrun [0] (in _CAN_RFR)
- #define _CAN_RFOM ((uint8_t) (0x01 << 5))

  CAN Release FIFO Output Mailbox [0] (in _CAN_RFR)
- #define _CAN_TMEIE ((uint8_t) (0x01 << 0))

  CAN Transmit Mailbox Empty Interrupt Enable [0] (in _CAN_IER)
- #define _CAN_FMPIE ((uint8_t) (0x01 << 1))

  CAN FIFO Message Pending Interrupt Enable [0] (in _CAN_IER)
- #define _CAN_FFIE ((uint8_t) (0x01 << 2))

  CAN FIFO Full Interrupt Enable [0] (in _CAN_IER)
- #define _CAN_FOVIE ((uint8_t) (0x01 << 3))

  CAN FIFO Overrun Interrupt Enable [0] (in _CAN_IER)
- #define _CAN_WKUIE ((uint8_t) (0x01 << 7))

  CAN Wakeup Interrupt Enable [0] (in _CAN_IER)
- #define _CAN_LBKM ((uint8_t) (0x01 << 0))

  CAN Loop back mode [0] (in _CAN_DGR)
- #define _CAN_SILM ((uint8_t) (0x01 << 1))

  CAN Silent mode [0] (in _CAN_DGR)
- #define _CAN_SAMP ((uint8_t) (0x01 << 2))

  CAN Last sample point [0] (in _CAN_DGR)

- #define _CAN_RXS ((uint8_t) (0x01 << 3))
  CAN Rx Signal (=pin status) [0] (in _CAN_DGR)
- #define _CAN_TXM2E ((uint8_t) (0x01 << 4))
  CAN TX Mailbox 2 enable [0] (in _CAN_DGR)
- #define _CAN_PS ((uint8_t) (0x07 << 0))
  CAN Page select [2:0] (in _CAN_PSR)
- #define _CAN_PS0 ((uint8_t) (0x01 << 0))
  CAN Page select [0] (in _CAN_PSR)
- #define _CAN_PS1 ((uint8_t) (0x01 << 1))
  CAN Page select [1] (in _CAN_PSR)
- #define _CAN_PS2 ((uint8_t) (0x01 << 2))
  CAN Page select [2] (in _CAN_PSR)
- #define _CAN_TXRQ ((uint8_t) (0x01 << 0))
  CAN Transmission mailbox request [0] (in _CAN_MCSR, page 0,1,5)
- #define _CAN_ABRQ ((uint8_t) (0x01 << 1))
  CAN Abort request for mailbox [0] (in _CAN_MCSR, page 0,1,5)
- #define _CAN_RQCP ((uint8_t) (0x01 << 2))
  CAN Request completed [0] (in _CAN_MCSR, page 0,1,5)
- #define _CAN_TXOK ((uint8_t) (0x01 << 3))
  CAN Transmission OK [0] (in _CAN_MCSR, page 0,1,5)
- #define _CAN_ALST ((uint8_t) (0x01 << 4))
  CAN Arbitration lost [0] (in _CAN_MCSR, page 0,1,5)
- #define _CAN_TERR ((uint8_t) (0x01 << 5))
  CAN Transmission error [0] (in _CAN_MCSR, page 0,1,5)
- #define _CAN_DLC ((uint8_t) (0x0F << 0))
  CAN Data length code [3:0] (in _CAN_MDLCR, page 0,1,5,7)
- #define _CAN_DLC0 ((uint8_t) (0x01 << 0))
  CAN Data length code [0] (in _CAN_MDLCR, page 0,1,5,7)
- #define _CAN_DLC1 ((uint8_t) (0x01 << 1))
  CAN Data length code [1] (in _CAN_MDLCR, page 0,1,5,7)
- #define _CAN_DLC2 ((uint8_t) (0x01 << 2))
  CAN Data length code [2] (in _CAN_MDLCR, page 0,1,5,7)
- #define _CAN_DLC3 ((uint8_t) (0x01 << 3))
  CAN Data length code [3] (in _CAN_MDLCR, page 0,1,5,7)
- #define _CAN_TGT ((uint8_t) (0x01 << 7))
  CAN Transmit global time [0] (in _CAN_MDLCR, page 0,1,5,7)
- #define _CAN_RTR ((uint8_t) (0x01 << 5))
  CAN Remote transmission request [0] (in _CAN_MIDR1, page 0,1,5)
- #define _CAN_IDE ((uint8_t) (0x01 << 6))
  CAN Extended identifier [0] (in _CAN_MIDR1, page 0,1,5)
- #define _CAN_EWGF ((uint8_t) (0x01 << 0))
  CAN Error warning flag [0] (in _CAN_ESR, page 6)
- #define _CAN_EPVF ((uint8_t) (0x01 << 1))
  CAN Error passive flag [0] (in _CAN_ESR, page 6)
- #define _CAN_BOFF ((uint8_t) (0x01 << 2))
  CAN Bus off flag [0] (in _CAN_ESR, page 6)
- #define _CAN_LEC ((uint8_t) (0x07 << 4))
  CAN Last error code [2:0] (in _CAN_ESR, page 6)
- #define _CAN_LEC0 ((uint8_t) (0x01 << 4))
  CAN Last error code [0] (in _CAN_ESR, page 6)
- #define _CAN_LEC1 ((uint8_t) (0x01 << 5))

CAN Last error code [1] (in _CAN_ESR, page 6)

- #define _CAN_LEC2 ((uint8_t) (0x01 << 6))

  CAN Last error code [3] (in _CAN_ESR, page 6)
- #define _CAN_EWGIE ((uint8_t) (0x01 << 0))

  CAN Error warning interrupt enable [0] (in _CAN_EIER, page 6)
- #define _CAN_EPVIE ((uint8_t) (0x01 << 1))

  CAN Error passive interrupt enable [0] (in _CAN_EIER, page 6)
- #define _CAN_BOFIE ((uint8_t) (0x01 << 2))

  CAN Bus-Off interrupt enable [0] (in _CAN_EIER, page 6)
- #define _CAN_LECIE ((uint8_t) (0x01 << 4))

  CAN Last error code interrupt enable [0] (in _CAN_EIER, page 6)
- #define _CAN_ERRIE ((uint8_t) (0x01 << 6))

  CAN Error interrupt enable [0] (in _CAN_EIER, page 6)
- #define _CAN_BRP ((uint8_t) (0x3F << 0))

  CAN Baud rate prescaler [5:0] (in _CAN_BTR1, page 6)
- #define _CAN_BRP0 ((uint8_t) (0x01 << 0))

  CAN Baud rate prescaler [0] (in _CAN_BTR1, page 6)
- #define _CAN_BRP1 ((uint8_t) (0x01 << 1))

  CAN Baud rate prescaler [1] (in _CAN_BTR1, page 6)
- #define _CAN_BRP2 ((uint8_t) (0x01 << 2))

  CAN Baud rate prescaler [2] (in _CAN_BTR1, page 6)
- #define _CAN_BRP3 ((uint8_t) (0x01 << 3))

  CAN Baud rate prescaler [3] (in _CAN_BTR1, page 6)
- #define _CAN_BRP4 ((uint8_t) (0x01 << 4))

  CAN Baud rate prescaler [4] (in _CAN_BTR1, page 6)
- #define _CAN_BRP5 ((uint8_t) (0x01 << 5))

  CAN Baud rate prescaler [5] (in _CAN_BTR1, page 6)
- #define _CAN_SJW ((uint8_t) (0x03 << 6))

  CAN Resynchronization jump width [1:0] (in _CAN_EIER, page 6)
- #define _CAN_SJW0 ((uint8_t) (0x01 << 6))

  CAN Resynchronization jump width [0] (in _CAN_EIER, page 6)
- #define _CAN_SJW1 ((uint8_t) (0x01 << 7))

  CAN Resynchronization jump width [1] (in _CAN_EIER, page 6)
- #define _CAN_BS1 ((uint8_t) (0x0F << 0))

  CAN Bit segment 1 [3:0] (in _CAN_BTR2, page 6)
- #define _CAN_BS10 ((uint8_t) (0x01 << 0))

  CAN Bit segment 1 [0] (in _CAN_BTR2, page 6)
- #define _CAN_BS11 ((uint8_t) (0x01 << 1))

  CAN Bit segment 1 [1] (in _CAN_BTR2, page 6)
- #define _CAN_BS12 ((uint8_t) (0x01 << 2))

  CAN Bit segment 1 [2] (in _CAN_BTR2, page 6)
- #define _CAN_BS13 ((uint8_t) (0x01 << 3))

  CAN Bit segment 1 [3] (in _CAN_BTR2, page 6)
- #define _CAN_BS2 ((uint8_t) (0x07 << 4))

  CAN Bit segment 2 [2:0] (in _CAN_BTR2, page 6)
- #define _CAN_BS20 ((uint8_t) (0x01 << 4))

  CAN Bit segment 2 [0] (in _CAN_BTR2, page 6)
- #define _CAN_BS21 ((uint8_t) (0x01 << 5))

  CAN Bit segment 2 [1] (in _CAN_BTR2, page 6)
- #define _CAN_BS22 ((uint8_t) (0x01 << 6))

  CAN Bit segment 2 [2] (in _CAN_BTR2, page 6)

- #define _CAN_FML0 ((uint8_t) (0x01 << 0))
  CAN Filter 0 mode low [0] (in _CAN_FMR1, page 6)
- #define _CAN_FMH0 ((uint8_t) (0x01 << 1))
  CAN Filter 0 mode high [0] (in _CAN_FMR1, page 6)
- #define _CAN_FML1 ((uint8_t) (0x01 << 2))
  CAN Filter 1 mode low [0] (in _CAN_FMR1, page 6)
- #define _CAN_FMH1 ((uint8_t) (0x01 << 3))
  CAN Filter 1 mode high [0] (in _CAN_FMR1, page 6)
- #define _CAN_FML2 ((uint8_t) (0x01 << 4))
  CAN Filter 2 mode low [0] (in _CAN_FMR1, page 6)
- #define _CAN_FMH2 ((uint8_t) (0x01 << 5))
  CAN Filter 2 mode high [0] (in _CAN_FMR1, page 6)
- #define _CAN_FML3 ((uint8_t) (0x01 << 6))
  CAN Filter 3 mode low [0] (in _CAN_FMR1, page 6)
- #define _CAN_FMH3 ((uint8_t) (0x01 << 7))
  CAN Filter 3 mode high [0] (in _CAN_FMR1, page 6)
- #define _CAN_FML4 ((uint8_t) (0x01 << 0))
  CAN Filter 4 mode low [0] (in _CAN_FMR2, page 6)
- #define _CAN_FMH4 ((uint8_t) (0x01 << 1))
  CAN Filter 4 mode high [0] (in _CAN_FMR2, page 6)
- #define _CAN_FML5 ((uint8_t) (0x01 << 2))
  CAN Filter 5 mode low [0] (in _CAN_FMR2, page 6)
- #define _CAN_FMH5 ((uint8_t) (0x01 << 3))
  CAN Filter 5 mode high [0] (in _CAN_FMR2, page 6)
- #define _CAN_FACT0 ((uint8_t) (0x01 << 0))
  CAN Filter 0 active [0] (in _CAN_FCR1, page 6)
- #define _CAN_FSC0 ((uint8_t) (0x03 << 1))
  CAN Filter 0 scale configuration [1:0] (in _CAN_FCR1, page 6)
- #define _CAN_FSC00 ((uint8_t) (0x01 << 1))
  CAN Filter 0 scale configuration [0] (in _CAN_FCR1, page 6)
- #define _CAN_FSC01 ((uint8_t) (0x01 << 2))
  CAN Filter 0 scale configuration [1] (in _CAN_FCR1, page 6)
- #define _CAN_FACT1 ((uint8_t) (0x01 << 4))
  CAN Filter 1 active [0] (in _CAN_FCR1, page 6)
- #define _CAN_FSC1 ((uint8_t) (0x03 << 5))
  CAN Filter 1 scale configuration [1:0] (in _CAN_FCR1, page 6)
- #define _CAN_FSC10 ((uint8_t) (0x01 << 5))
  CAN Filter 1 scale configuration [0] (in _CAN_FCR1, page 6)
- #define _CAN_FSC11 ((uint8_t) (0x01 << 6))
  CAN Filter 1 scale configuration [1] (in _CAN_FCR1, page 6)
- #define _CAN_FACT2 ((uint8_t) (0x01 << 0))
  CAN Filter 2 active [0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC2 ((uint8_t) (0x03 << 1))
  CAN Filter 2 scale configuration [1:0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC20 ((uint8_t) (0x01 << 1))
  CAN Filter 2 scale configuration [0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC21 ((uint8_t) (0x01 << 2))
  CAN Filter 2 scale configuration [1] (in _CAN_FCR2, page 6)
- #define _CAN_FACT3 ((uint8_t) (0x01 << 4))
  CAN Filter 3 active [0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC3 ((uint8_t) (0x03 << 5))

CAN Filter 3 scale configuration [1:0] (in _CAN_FCR2, page 6)

- #define _CAN_FSC30 ((uint8_t) (0x01 << 5))

  CAN Filter 3 scale configuration [0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC31 ((uint8_t) (0x01 << 6))

  CAN Filter 3 scale configuration [1] (in _CAN_FCR2, page 6)
- #define _CAN_FACT4 ((uint8_t) (0x01 << 0))

  CAN Filter 4 active [0] (in _CAN_FCR3, page 6)
- #define _CAN_FSC4 ((uint8_t) (0x03 << 1))

  CAN Filter 4 scale configuration [1:0] (in _CAN_FCR3, page 6)
- #define _CAN_FSC40 ((uint8_t) (0x01 << 1))

  CAN Filter 4 scale configuration [0] (in _CAN_FCR3, page 6)
- #define _CAN_FSC41 ((uint8_t) (0x01 << 2))

  CAN Filter 4 scale configuration [1] (in _CAN_FCR3, page 6)
- #define _CAN_FACT5 ((uint8_t) (0x01 << 4))

  CAN Filter 5 active [0] (in _CAN_FCR2, page 6)
- #define _CAN_FSC5 ((uint8_t) (0x03 << 5))

  CAN Filter 5 scale configuration [1:0] (in _CAN_FCR3, page 6)
- #define _CAN_FSC50 ((uint8_t) (0x01 << 5))

  CAN Filter 5 scale configuration [0] (in _CAN_FCR3, page 6)
- #define _CAN_FSC51 ((uint8_t) (0x01 << 6))

  CAN Filter 5 scale configuration [1] (in _CAN_FCR3, page 6)
- #define _CFG _SFR(CFG_t, CFG_AddressBase)

  CFG struct/bit access.
- #define _CFG_GCR _SFR(uint8_t, CFG_AddressBase+0x00)

  Global configuration register (CFG_GCR)
- #define _CFG_GCR_RESET_VALUE ((uint8_t)0x00)
- #define _CFG_SWD ((uint8_t) (0x01 << 0))

  SWIM disable [0].
- #define _CFG_AL ((uint8_t) (0x01 << 1))

  Activation level [0].
- #define _ITC _SFR(ITC_t, ITC_AddressBase)

  ITC struct/bit access.
- #define _ITC_SPR1 _SFR(uint8_t, ITC_AddressBase+0x00)

  Interrupt priority register 1/8.
- #define _ITC_SPR2 _SFR(uint8_t, ITC_AddressBase+0x01)

  Interrupt priority register 2/8.
- #define _ITC_SPR3 _SFR(uint8_t, ITC_AddressBase+0x02)

  Interrupt priority register 3/8.
- #define _ITC_SPR4 _SFR(uint8_t, ITC_AddressBase+0x03)

  Interrupt priority register 4/8.
- #define _ITC_SPR5 _SFR(uint8_t, ITC_AddressBase+0x04)

  Interrupt priority register 5/8.
- #define _ITC_SPR6 _SFR(uint8_t, ITC_AddressBase+0x05)

  Interrupt priority register 6/8.
- #define _ITC_SPR7 _SFR(uint8_t, ITC_AddressBase+0x06)

  Interrupt priority register 7/8.
- #define _ITC_SPR8 _SFR(uint8_t, ITC_AddressBase+0x07)

  Interrupt priority register 8/8.
- #define _ITC_SPR1_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 1/8 reset value.
- #define _ITC_SPR2_RESET_VALUE ((uint8_t) 0xFF)

Interrupt priority register 2/8 reset value.

- #define _ITC_SPR3_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 3/8 reset value.
- #define _ITC_SPR4_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 4/8 reset value.
- #define _ITC_SPR5_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 5/8 reset value.
- #define _ITC_SPR6_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 6/8 reset value.
- #define _ITC_SPR7_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 7/8 reset value.
- #define _ITC_SPR8_RESET_VALUE ((uint8_t) 0x0F)

  Interrupt priority register 8/8 reset value.
- #define _ITC_VECT1SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 1 [1:0] (in _ITC_SPR1)
- #define _ITC_VECT1SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 1 [0] (in _ITC_SPR1)
- #define _ITC_VECT1SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 1 [1] (in _ITC_SPR1)
- #define _ITC_VECT2SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 2 [1:0] (in _ITC_SPR1)
- #define _ITC_VECT2SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 2 [0] (in _ITC_SPR1)
- #define _ITC_VECT2SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 2 [1] (in _ITC_SPR1)
- #define _ITC_VECT3SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 3 [1:0] (in _ITC_SPR1)
- #define _ITC_VECT3SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 3 [0] (in _ITC_SPR1)
- #define _ITC_VECT3SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 3 [1] (in _ITC_SPR1)
- #define _ITC_VECT4SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 4 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT4SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 4 [0] (in _ITC_SPR2)
- #define _ITC_VECT4SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 4 [1] (in _ITC_SPR2)
- #define _ITC_VECT5SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 5 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT5SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 5 [0] (in _ITC_SPR2)
- #define _ITC_VECT5SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 5 [1] (in _ITC_SPR2)
- #define _ITC_VECT6SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 6 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT6SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 6 [0] (in _ITC_SPR2)
- #define _ITC_VECT6SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 6 [1] (in _ITC_SPR2)
- #define _ITC_VECT7SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 7 [1:0] (in _ITC_SPR2)

- #define _ITC_VECT7SPR0 ((uint8_t) (0x01 << 6))
  ITC interrupt priority vector 7 [0] (in _ITC_SPR2)
- #define _ITC_VECT7SPR1 ((uint8_t) (0x01 << 7))
  ITC interrupt priority vector 7 [1] (in _ITC_SPR2)
- #define _ITC_VECT8SPR ((uint8_t) (0x03 << 0))
  ITC interrupt priority vector 8 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT8SPR0 ((uint8_t) (0x01 << 0))
  ITC interrupt priority vector 8 [0] (in _ITC_SPR3)
- #define _ITC_VECT8SPR1 ((uint8_t) (0x01 << 1))
  ITC interrupt priority vector 8 [1] (in _ITC_SPR3)
- #define _ITC_VECT9SPR ((uint8_t) (0x03 << 2))
  ITC interrupt priority vector 9 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT9SPR0 ((uint8_t) (0x01 << 2))
  ITC interrupt priority vector 9 [0] (in _ITC_SPR3)
- #define _ITC_VECT9SPR1 ((uint8_t) (0x01 << 3))
  ITC interrupt priority vector 9 [1] (in _ITC_SPR3)
- #define _ITC_VECT10SPR ((uint8_t) (0x03 << 4))
  ITC interrupt priority vector 10 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT10SPR0 ((uint8_t) (0x01 << 4))
  ITC interrupt priority vector 10 [0] (in _ITC_SPR3)
- #define _ITC_VECT10SPR1 ((uint8_t) (0x01 << 5))
  ITC interrupt priority vector 10 [1] (in _ITC_SPR3)
- #define _ITC_VECT11SPR ((uint8_t) (0x03 << 6))
  ITC interrupt priority vector 11 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT11SPR0 ((uint8_t) (0x01 << 6))
  ITC interrupt priority vector 11 [0] (in _ITC_SPR3)
- #define _ITC_VECT11SPR1 ((uint8_t) (0x01 << 7))
  ITC interrupt priority vector 11 [1] (in _ITC_SPR3)
- #define _ITC_VECT12SPR ((uint8_t) (0x03 << 0))
  ITC interrupt priority vector 12 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT12SPR0 ((uint8_t) (0x01 << 0))
  ITC interrupt priority vector 12 [0] (in _ITC_SPR4)
- #define _ITC_VECT12SPR1 ((uint8_t) (0x01 << 1))
  ITC interrupt priority vector 12 [1] (in _ITC_SPR4)
- #define _ITC_VECT13SPR ((uint8_t) (0x03 << 2))
  ITC interrupt priority vector 13 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT13SPR0 ((uint8_t) (0x01 << 2))
  ITC interrupt priority vector 13 [0] (in _ITC_SPR4)
- #define _ITC_VECT13SPR1 ((uint8_t) (0x01 << 3))
  ITC interrupt priority vector 13 [1] (in _ITC_SPR4)
- #define _ITC_VECT14SPR ((uint8_t) (0x03 << 4))
  ITC interrupt priority vector 14 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT14SPR0 ((uint8_t) (0x01 << 4))
  ITC interrupt priority vector 14 [0] (in _ITC_SPR4)
- #define _ITC_VECT14SPR1 ((uint8_t) (0x01 << 5))
  ITC interrupt priority vector 14 [1] (in _ITC_SPR4)
- #define _ITC_VECT15SPR ((uint8_t) (0x03 << 6))
  ITC interrupt priority vector 15 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT15SPR0 ((uint8_t) (0x01 << 6))
  ITC interrupt priority vector 15 [0] (in _ITC_SPR4)
- #define _ITC_VECT15SPR1 ((uint8_t) (0x01 << 7))

ITC interrupt priority vector 15 [1] (in _ITC_SPR4)

- #define _ITC_VECT16SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 16 [1:0] (in _ITC_SPR5)
- #define _ITC_VECT16SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 16 [0] (in _ITC_SPR5)
- #define _ITC_VECT16SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 16 [1] (in _ITC_SPR5)
- #define _ITC_VECT17SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 17 [1:0] (in _ITC_SPR5)
- #define _ITC_VECT17SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 17 [0] (in _ITC_SPR5)
- #define _ITC_VECT17SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 17 [1] (in _ITC_SPR5)
- #define _ITC_VECT18SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 18 [1:0] (in _ITC_SPR5)
- #define _ITC_VECT18SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 18 [0] (in _ITC_SPR5)
- #define _ITC_VECT18SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 18 [1] (in _ITC_SPR5)
- #define _ITC_VECT19SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 19 [1:0] (in _ITC_SPR5)
- #define _ITC_VECT19SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 19 [0] (in _ITC_SPR5)
- #define _ITC_VECT19SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 19 [1] (in _ITC_SPR5)
- #define _ITC_VECT20SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 20 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT20SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 20 [0] (in _ITC_SPR6)
- #define _ITC_VECT20SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 20 [1] (in _ITC_SPR6)
- #define _ITC_VECT21SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 21 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT21SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 21 [0] (in _ITC_SPR6)
- #define _ITC_VECT21SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 21 [1] (in _ITC_SPR6)
- #define _ITC_VECT22SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 22 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT22SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 22 [0] (in _ITC_SPR6)
- #define _ITC_VECT22SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 22 [1] (in _ITC_SPR6)
- #define _ITC_VECT23SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 23 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT23SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 23 [0] (in _ITC_SPR6)
- #define _ITC_VECT23SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 23 [1] (in _ITC_SPR6)
- #define _ITC_VECT24SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 24 [1:0] (in _ITC_SPR7)

- #define _ITC_VECT24SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 24 [0] (in _ITC_SPR7)
- #define _ITC_VECT24SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 24 [1] (in _ITC_SPR7)
- #define _ITC_VECT25SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 25 [1:0] (in _ITC_SPR7)
- #define _ITC_VECT25SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 25 [0] (in _ITC_SPR7)
- #define _ITC_VECT25SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 25 [1] (in _ITC_SPR7)
- #define _ITC_VECT26SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 26 [1:0] (in _ITC_SPR7)
- #define _ITC_VECT26SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 26 [0] (in _ITC_SPR7)
- #define _ITC_VECT26SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 26 [1] (in _ITC_SPR7)
- #define _ITC_VECT27SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 27 [1:0] (in _ITC_SPR7)
- #define _ITC_VECT27SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 27 [0] (in _ITC_SPR7)
- #define _ITC_VECT27SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 27 [1] (in _ITC_SPR7)
- #define _ITC_VECT28SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 28 [1:0] (in _ITC_SPR8)
- #define _ITC_VECT28SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 28 [0] (in _ITC_SPR8)
- #define _ITC_VECT28SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 28 [1] (in _ITC_SPR8)
- #define _ITC_VECT29SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 29 [1:0] (in _ITC_SPR8)
- #define _ITC_VECT29SPR0 ((uint8_t) (0x01 << 2))

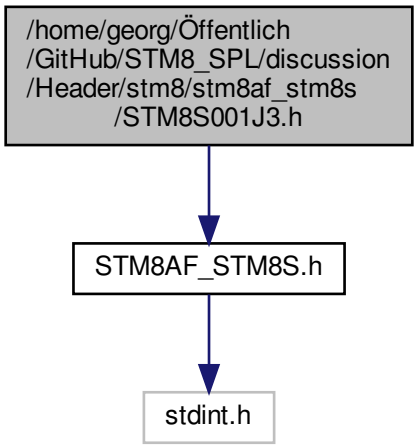
  ITC interrupt priority vector 29 [0] (in _ITC_SPR8)
- #define _ITC_VECT29SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 29 [1] (in _ITC_SPR8)

7.33 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S001J3.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S001J3.h:



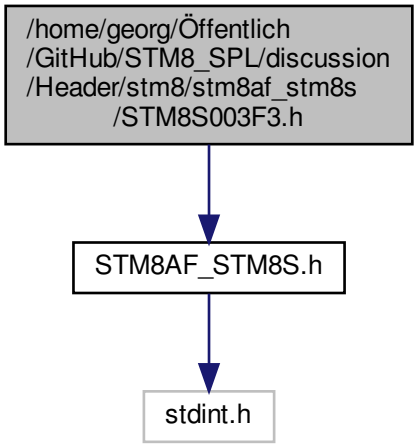
Macros

- #define STM8S001J3
- #define STM8S001
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 128
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.34 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S003F3.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S003F3.h:



Macros

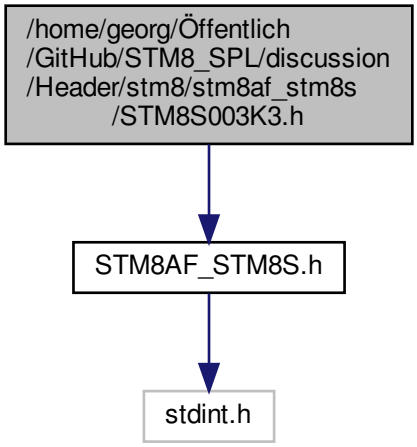
- #define STM8S003F3
- #define STM8S003
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 128
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230

- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.35 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S003K3.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S003K3.h:



Macros

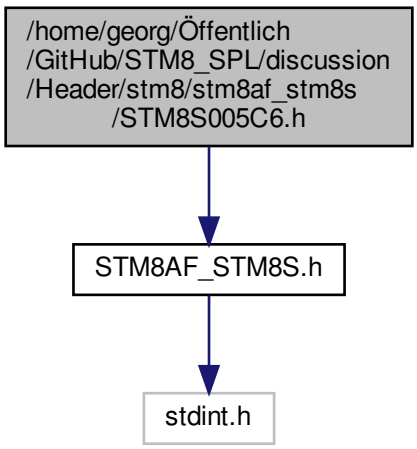
- #define STM8S003K3
- #define STM8S003
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 128
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0

- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.36 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S005C6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S005C6.h:



Macros

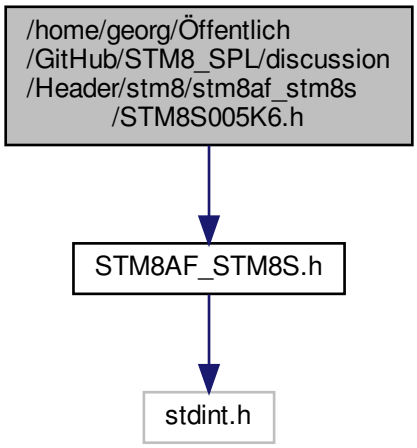
- #define STM8S005C6
- #define STM8S005
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 128

- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.37 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S005K6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S005K6.h:



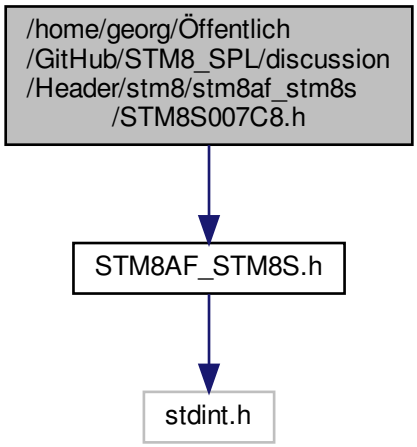
Macros

- #define STM8S005K6
- #define STM8S005
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 128
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.38 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S007C8.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S007C8.h:



Macros

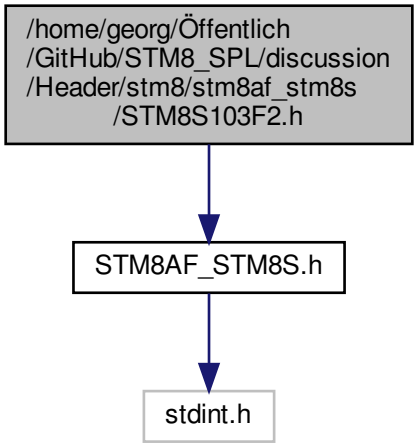
- #define STM8S007C8
- #define STM8S007
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 128
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90

7.39 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S103F2.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S103F2.h:



Macros

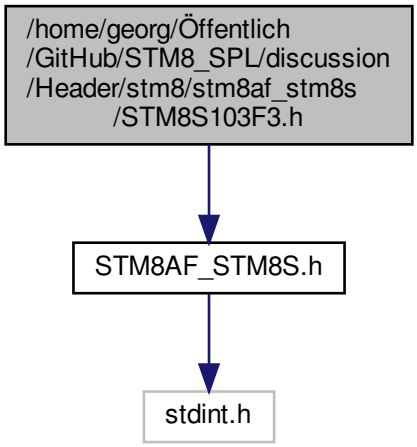
- #define STM8S103F2
- #define STM8S103
- #define STM8_PFLASH_SIZE 4096
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0

- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4865

7.40 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S103F3.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S103F3.h:



Macros

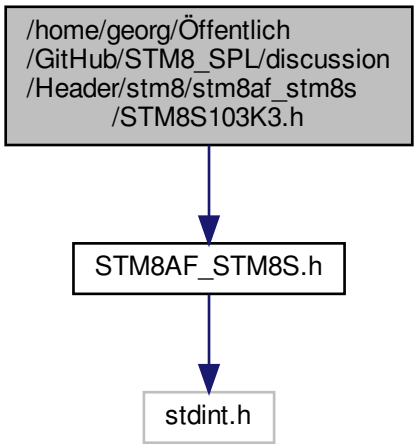
- #define STM8S103F3
- #define STM8S103
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024

- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4865

7.41 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S103K3.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S103K3.h:



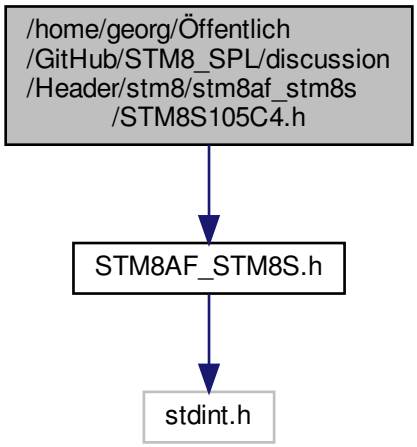
Macros

- #define STM8S103K3
- #define STM8S103
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4865

7.42 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S105C4.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S105C4.h:



Macros

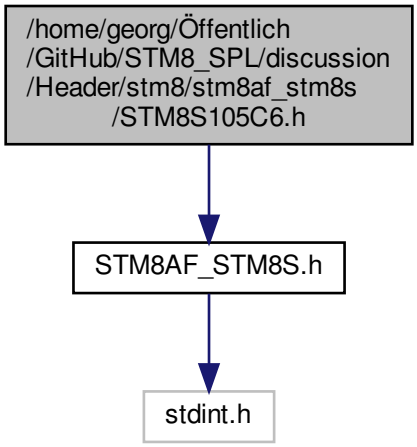
- #define STM8S105C4
- #define STM8S105
- #define STM8_PFLASH_SIZE 16384
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340

- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.43 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S105C6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S105C6.h:



Macros

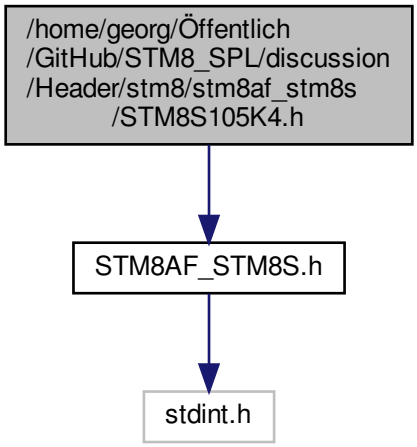
- #define STM8S105C6
- #define STM8S105
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3

- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.44 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S105K4.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S105K4.h:



Macros

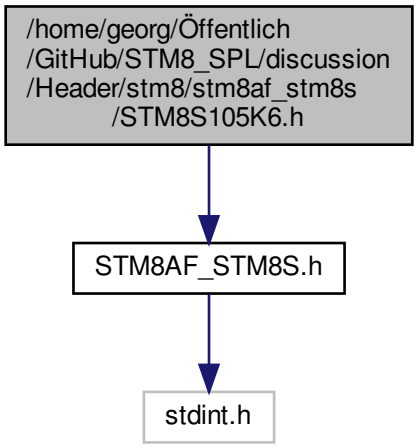
- #define STM8S105K4
- #define STM8S105
- #define STM8_PFLASH_SIZE 16384
- #define STM8_RAM_SIZE 2048

- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.45 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S105K6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S105K6.h:



Macros

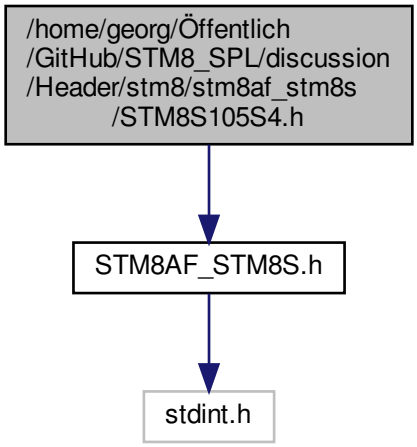
- #define STM8S105K6
- #define STM8S105
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340

- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.46 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S105S4.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S105S4.h:



Macros

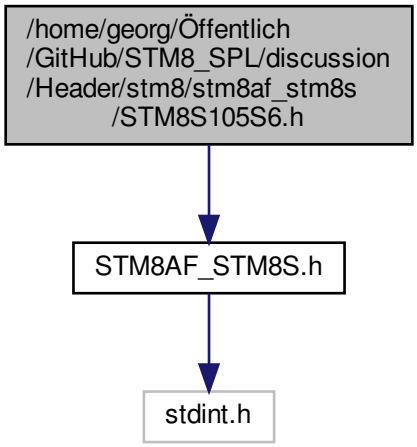
- #define STM8S105S4
- #define STM8S105
- #define STM8_PFLASH_SIZE 16384
- #define STM8_RAM_SIZE 2048
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3

- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.47 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S105S6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S105S6.h:



Macros

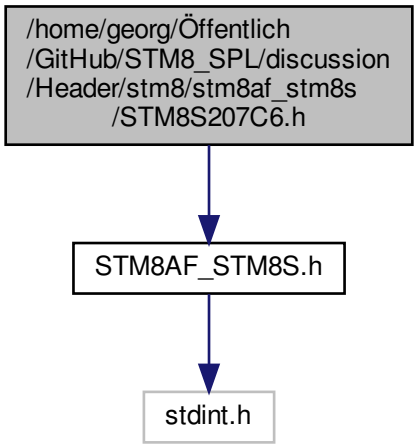
- #define STM8S105S6
- #define STM8S105
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 2048

- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART2_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.48 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S207C6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207C6.h:



Macros

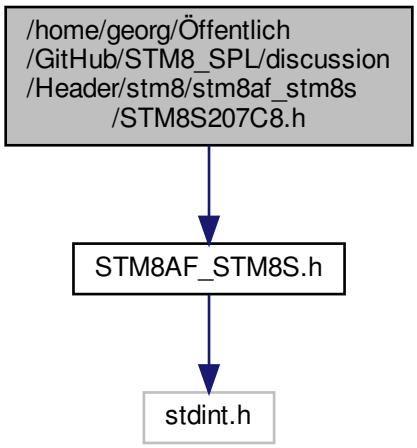
- #define STM8S207C6
- #define STM8S207
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.49 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S207C8.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207C8.h:



Macros

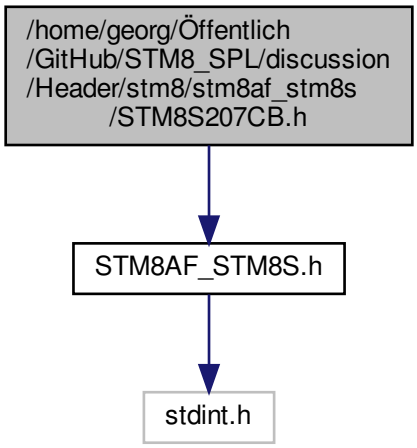
- #define STM8S207C8
- #define STM8S207
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E

- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.50 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S207CB.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207CB.h:



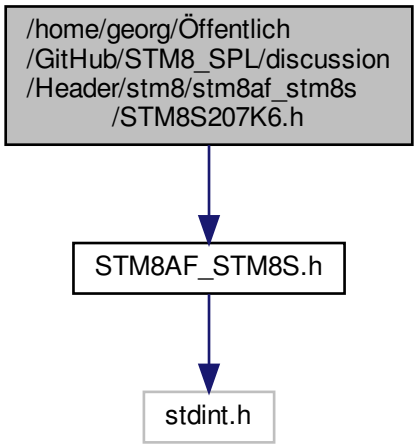
Macros

- #define STM8S207CB
- #define STM8S207
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.51 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/↩ STM8S207K6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207K6.h:



Macros

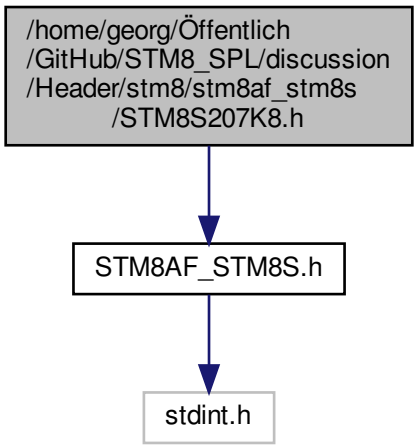
- #define STM8S207K6
- #define STM8S207
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.52 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S207K8.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207K8.h:



Macros

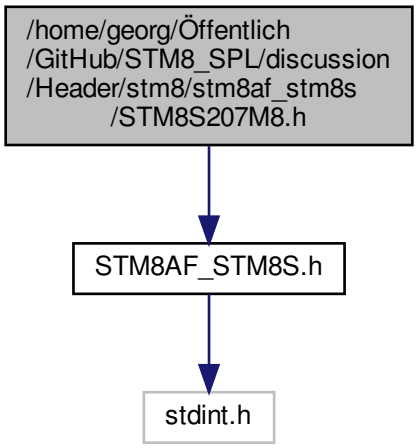
- #define STM8S207K8
- #define STM8S207
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E

- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.53 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S207M8.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207M8.h:



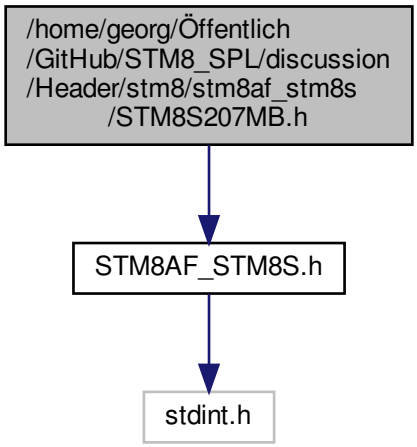
Macros

- #define STM8S207M8
- #define STM8S207
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.54 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S207MB.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207MB.h:



Macros

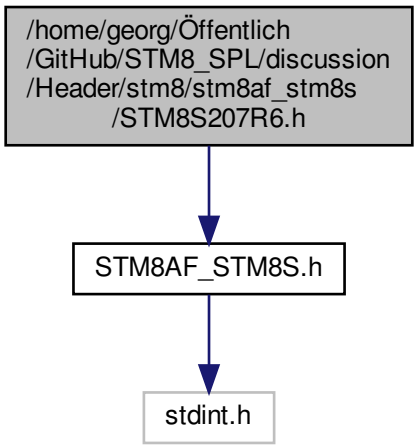
- #define STM8S207MB
- #define STM8S207
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.55 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S207R6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207R6.h:



Macros

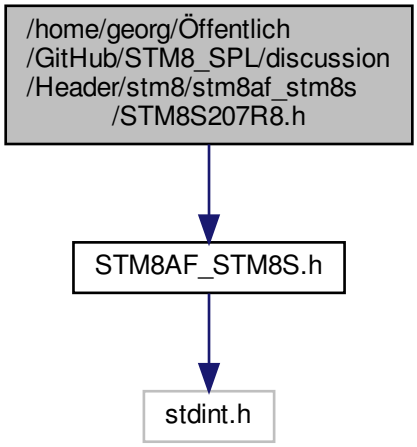
- #define STM8S207R6
- #define STM8S207
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E

- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.56 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S207R8.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207R8.h:



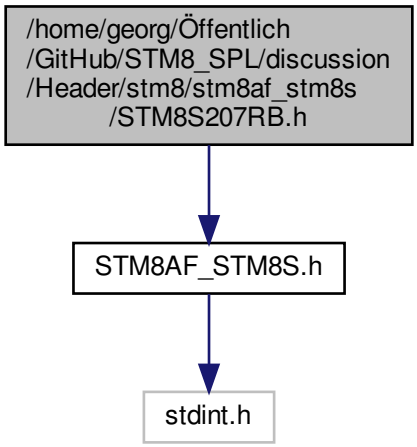
Macros

- #define STM8S207R8
- #define STM8S207
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.57 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/↩ STM8S207RB.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207RB.h:



Macros

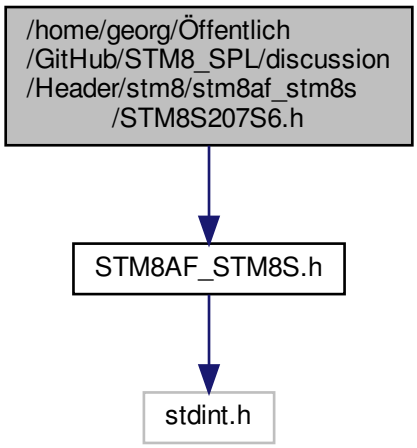
- #define STM8S207RB
- #define STM8S207
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.58 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S207S6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207S6.h:



Macros

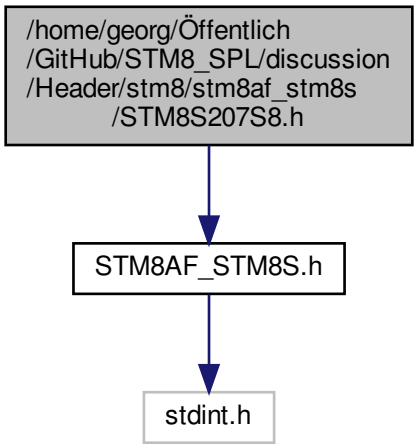
- #define STM8S207S6
- #define STM8S207
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1024
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E

- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.59 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S207S8.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207S8.h:



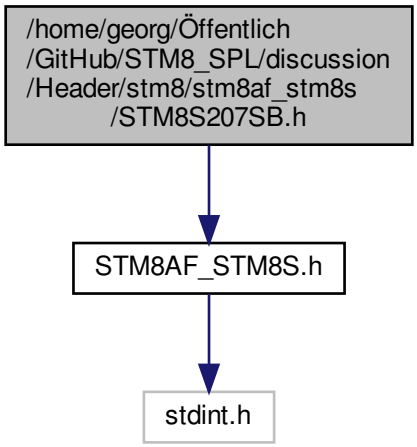
Macros

- #define STM8S207S8
- #define STM8S207
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.60 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/↩ STM8S207SB.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S207SB.h:



Macros

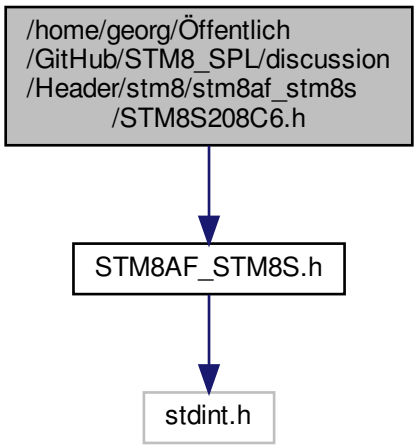
- #define STM8S207SB
- #define STM8S207
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.61 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S208C6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S208C6.h:



Macros

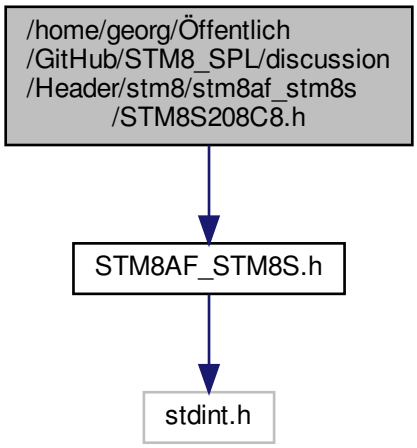
- #define STM8S208C6
- #define STM8S208
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E

- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.62 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S208C8.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S208C8.h:



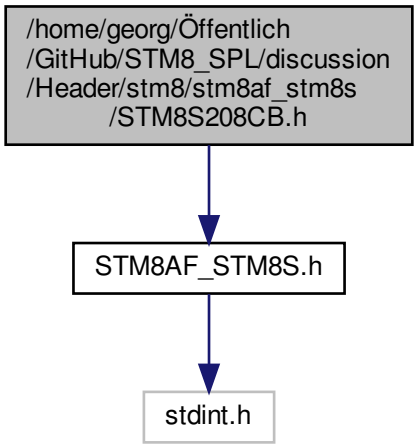
Macros

- #define STM8S208C8
- #define STM8S208
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.63 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S208CB.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S208CB.h:



Macros

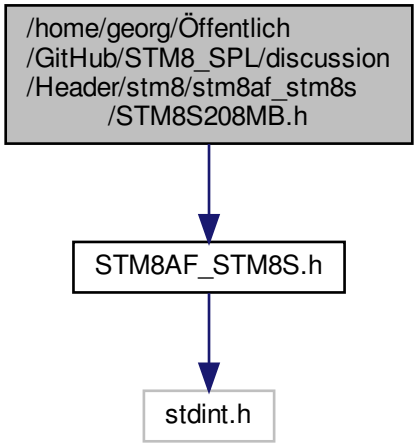
- #define STM8S208CB
- #define STM8S208
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.64 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S208MB.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S208MB.h:



Macros

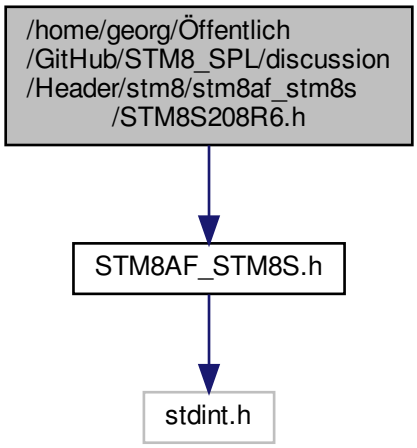
- #define STM8S208MB
- #define STM8S208
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019

- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.65 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S208R6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S208R6.h:



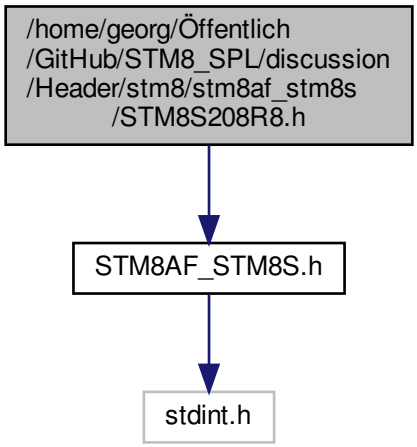
Macros

- #define STM8S208R6
- #define STM8S208
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.66 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S208R8.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S208R8.h:



Macros

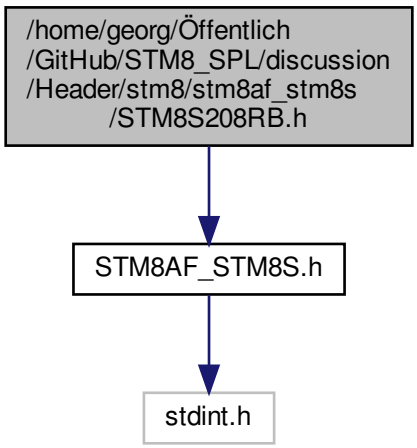
- #define STM8S208R8
- #define STM8S208
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.67 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S208RB.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S208RB.h:



Macros

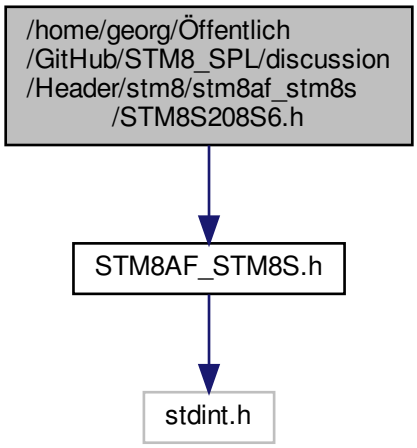
- #define STM8S208RB
- #define STM8S208
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 2048
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019

- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.68 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S208S6.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S208S6.h:



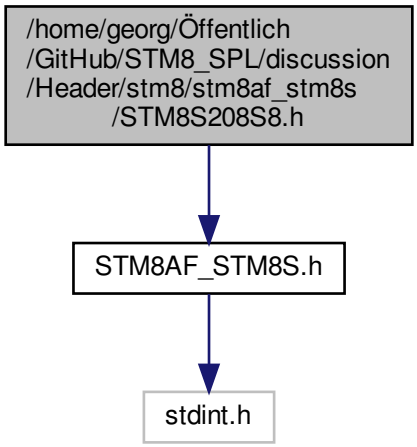
Macros

- #define STM8S208S6
- #define STM8S208
- #define STM8_PFLASH_SIZE 32768
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.69 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S208S8.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S208S8.h:



Macros

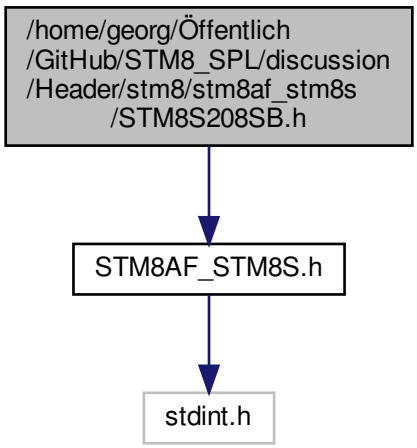
- #define STM8S208S8
- #define STM8S208
- #define STM8_PFLASH_SIZE 65536
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250

- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.70 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S208SB.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S208SB.h:



Macros

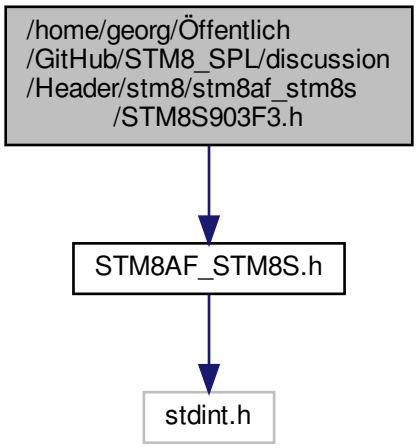
- #define STM8S208SB
- #define STM8S208
- #define STM8_PFLASH_SIZE 131072
- #define STM8_RAM_SIZE 6144
- #define STM8_EEPROM_SIZE 1536
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019

- #define PORTG_AddressBase 0x501E
- #define PORTH_AddressBase 0x5023
- #define PORTI_AddressBase 0x5028
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define UART3_AddressBase 0x5240
- #define TIM1_AddressBase 0x5250
- #define TIM2_AddressBase 0x5300
- #define TIM3_AddressBase 0x5320
- #define TIM4_AddressBase 0x5340
- #define ADC2_AddressBase 0x5400
- #define CAN_AddressBase 0x5420
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x48CD

7.71 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S903F3.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S903F3.h:



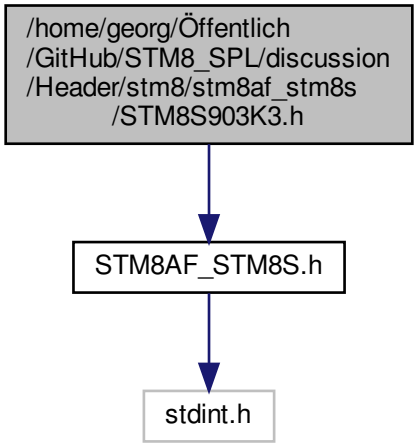
Macros

- #define STM8S903F3
- #define STM8S903
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4865

7.72 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8af_stm8s/STM8S903K3.h File Reference

```
#include "STM8AF_STM8S.h"
```

Include dependency graph for STM8S903K3.h:



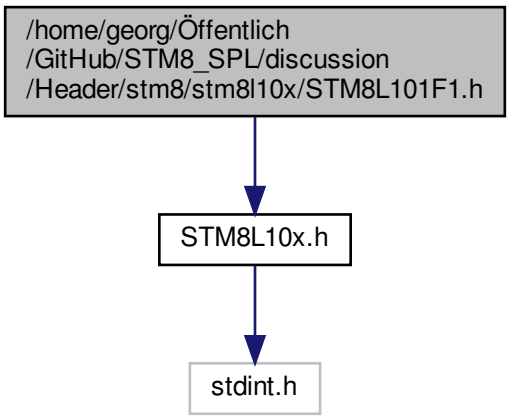
Macros

- #define STM8S903K3
- #define STM8S903
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1024
- #define STM8_EEPROM_SIZE 640
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define PORTE_AddressBase 0x5014
- #define PORTF_AddressBase 0x5019
- #define FLASH_AddressBase 0x505A
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B3
- #define CLK_AddressBase 0x50C0
- #define WWDG_AddressBase 0x50D1
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define UART1_AddressBase 0x5230
- #define TIM1_AddressBase 0x5250
- #define TIM5_AddressBase 0x5300
- #define TIM6_AddressBase 0x5340
- #define ADC1_AddressBase 0x53E0
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4865

7.73 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L101F1.h File Reference

```
#include "STM8L10x.h"
```

Include dependency graph for STM8L101F1.h:



Macros

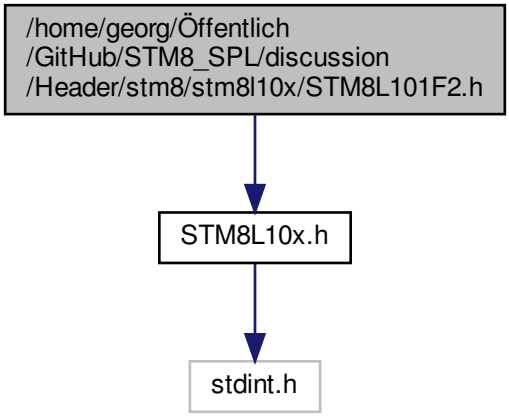
- #define STM8L101F1
- #define STM8L10x
- #define STM8_PFLASH_SIZE 2048
- #define STM8_RAM_SIZE 1536
- #define STM8_EEPROM_SIZE 0
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define FLASH_AddressBase 0x5050
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B0
- #define CLK_AddressBase 0x50C0
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define USART_AddressBase 0x5230
- #define WFE_AddressBase 0x50A6
- #define TIM2_AddressBase 0x5250
- #define TIM3_AddressBase 0x5280
- #define TIM4_AddressBase 0x52E0

- #define IRTIM_AddressBase 0x52FF
- #define COMP_AddressBase 0x5300
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4925

7.74 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L101F2.h File Reference

```
#include "STM8L10x.h"
```

Include dependency graph for STM8L101F2.h:



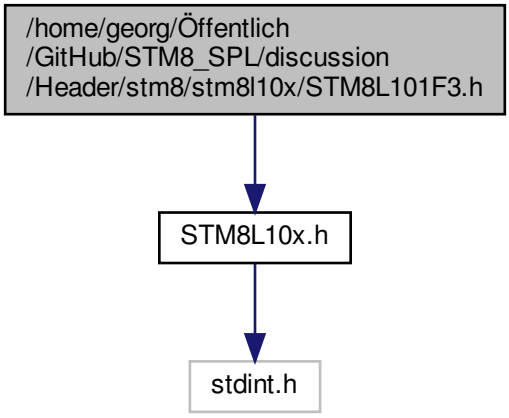
Macros

- #define STM8L101F2
- #define STM8L10x
- #define STM8_PFLASH_SIZE 4096
- #define STM8_RAM_SIZE 1536
- #define STM8_EEPROM_SIZE 0
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define FLASH_AddressBase 0x5050
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B0
- #define CLK_AddressBase 0x50C0
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0

- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define USART_AddressBase 0x5230
- #define WFE_AddressBase 0x50A6
- #define TIM2_AddressBase 0x5250
- #define TIM3_AddressBase 0x5280
- #define TIM4_AddressBase 0x52E0
- #define IRTIM_AddressBase 0x52FF
- #define COMP_AddressBase 0x5300
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4925

7.75 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L101F3.h File Reference

`#include "STM8L10x.h"`
Include dependency graph for STM8L101F3.h:



Macros

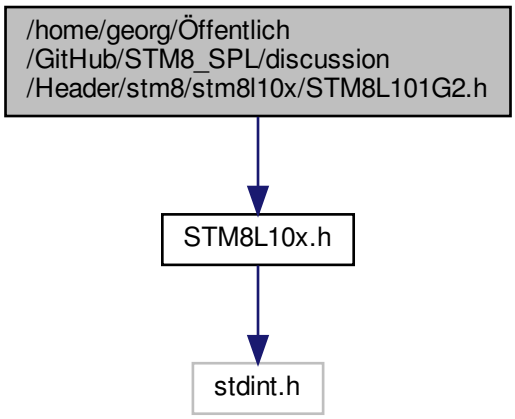
- #define STM8L101F3
- #define STM8L10x
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1536
- #define STM8_EEPROM_SIZE 0
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005

- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define FLASH_AddressBase 0x5050
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B0
- #define CLK_AddressBase 0x50C0
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define USART_AddressBase 0x5230
- #define WFE_AddressBase 0x50A6
- #define TIM2_AddressBase 0x5250
- #define TIM3_AddressBase 0x5280
- #define TIM4_AddressBase 0x52E0
- #define IRTIM_AddressBase 0x52FF
- #define COMP_AddressBase 0x5300
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4925

7.76 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L101G2.h File Reference

```
#include "STM8L10x.h"
```

Include dependency graph for STM8L101G2.h:



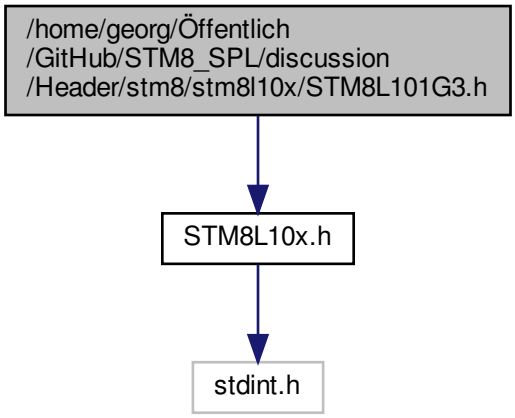
Macros

- #define STM8L101G2
- #define STM8L10x
- #define STM8_PFLASH_SIZE 4096
- #define STM8_RAM_SIZE 1536
- #define STM8_EEPROM_SIZE 0
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define FLASH_AddressBase 0x5050
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B0
- #define CLK_AddressBase 0x50C0
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define USART_AddressBase 0x5230
- #define WFE_AddressBase 0x50A6
- #define TIM2_AddressBase 0x5250
- #define TIM3_AddressBase 0x5280
- #define TIM4_AddressBase 0x52E0
- #define IRTIM_AddressBase 0x52FF
- #define COMP_AddressBase 0x5300
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4925

7.77 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L101G3.h File Reference

```
#include "STM8L10x.h"
```

Include dependency graph for STM8L101G3.h:



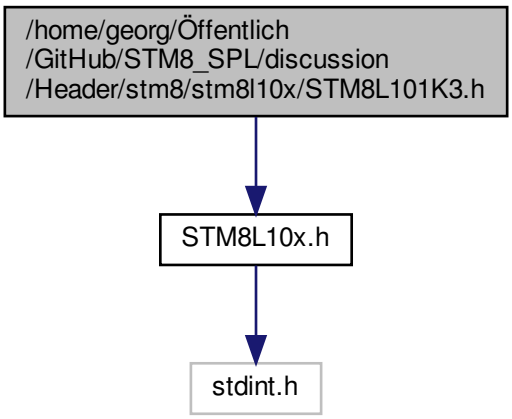
Macros

- #define STM8L101G3
- #define STM8L10x
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1536
- #define STM8_EEPROM_SIZE 0
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define FLASH_AddressBase 0x5050
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B0
- #define CLK_AddressBase 0x50C0
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define USART_AddressBase 0x5230
- #define WFE_AddressBase 0x50A6
- #define TIM2_AddressBase 0x5250
- #define TIM3_AddressBase 0x5280
- #define TIM4_AddressBase 0x52E0
- #define IRTIM_AddressBase 0x52FF
- #define COMP_AddressBase 0x5300
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4925

7.78 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L101K3.h File Reference

```
#include "STM8L10x.h"
```

Include dependency graph for STM8L101K3.h:



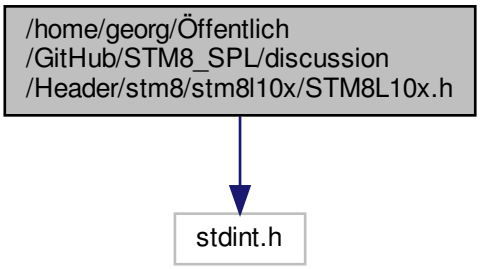
Macros

- #define STM8L101K3
- #define STM8L10x
- #define STM8_PFLASH_SIZE 8192
- #define STM8_RAM_SIZE 1536
- #define STM8_EEPROM_SIZE 0
- #define OPT_AddressBase 0x4800
- #define PORTA_AddressBase 0x5000
- #define PORTB_AddressBase 0x5005
- #define PORTC_AddressBase 0x500A
- #define PORTD_AddressBase 0x500F
- #define FLASH_AddressBase 0x5050
- #define EXTI_AddressBase 0x50A0
- #define RST_AddressBase 0x50B0
- #define CLK_AddressBase 0x50C0
- #define IWDG_AddressBase 0x50E0
- #define AWU_AddressBase 0x50F0
- #define BEEP_AddressBase 0x50F3
- #define SPI_AddressBase 0x5200
- #define I2C_AddressBase 0x5210
- #define USART_AddressBase 0x5230
- #define WFE_AddressBase 0x50A6
- #define TIM2_AddressBase 0x5250
- #define TIM3_AddressBase 0x5280
- #define TIM4_AddressBase 0x52E0

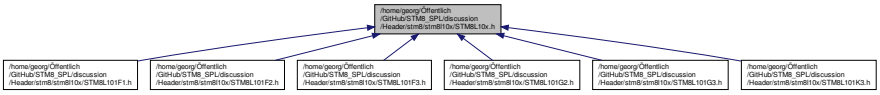
- #define IRTIM_AddressBase 0x52FF
- #define COMP_AddressBase 0x5300
- #define CFG_AddressBase 0x7F60
- #define ITC_AddressBase 0x7F70
- #define DM_AddressBase 0x7F90
- #define UID_AddressBase 0x4925

7.79 /home/georg/Öffentlich/GitHub/STM8_SPL/discussion/Header/stm8/stm8l10x/STM8L10x.h File Reference

`#include <stdint.h>`
Include dependency graph for STM8L10x.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct PORT_t
  structure for controlling pins in PORT mode (PORTx, x=A..I)
- struct FLASH_t
  struct to control write/erase of flash memory (FLASH)
- struct EXTI_t
  struct for configuring external port interrupts (EXTI)
- struct RST_t
  struct for determining reset source (RST)
- struct CLK_t
  struct for configuring/monitoring clock module (CLK)
- struct IWDG_t

struct for access to Independent Timeout Watchdog registers (IWDG)

- struct AWU_t

  struct for cofiguring the Auto Wake-Up Module (AWU)
- struct BEEP_t

  struct for beeper control (BEEP)
- struct SPI_t

  struct for controlling SPI module (SPI)
- struct I2C_t

  struct for controlling I2C module (I2C)
- struct USART_t

  struct for controlling Universal Asynchronous Receiver Transmitter (USART)
- struct WFE_t

  struct to configure interrupt sources as external interrupts or wake events (WFE)
- struct TIM2_3_t

  struct for controlling 16-Bit Timer 2+3 (TIM2, TIM3)
- struct TIM4_t

  struct for controlling 8-Bit Timer 4 (TIM4)
- struct IRTIM_t

  struct for Infrared Timer Module (IRTIM)
- struct COMP_t

  struct for Comparator Module (COMP)
- struct CFG_t

  struct for Global Configuration registers (CFG)
- struct ITC_t

  struct for setting interrupt Priority (ITC)

Macros

- #define STM8_PFLASH_SIZE 2048

  size of program flash [B]
- #define STM8_RAM_SIZE 1536

  size of RAM [B]
- #define STM8_EEPROM_SIZE 0

  size of data EEPROM [B]
- #define STM8_PFLASH_START 0x8000

  first address in program flash
- #define STM8_PFLASH_END (STM8_PFLASH_START + STM8_PFLASH_SIZE - 1)

  last address in program flash
- #define STM8_RAM_START 0x0000

  first address in RAM
- #define STM8_RAM_END (STM8_RAM_START + STM8_RAM_SIZE - 1)

  last address in RAM
- #define STM8_EEPROM_START 9800

  first address in EEPROM
- #define STM8_EEPROM_END (STM8_EEPROM_START + STM8_EEPROM_SIZE - 1)

  last address in EEPROM
- #define STM8_ADDR_WIDTH 16

  width of address space
- #define STM8_MEM_POINTER_T uint16_t

  address variable type

- #define ISR_HANDLER(func, irq) void func(void) __interrupt(irq)

  handler for interrupt service routine
- #define ISR_HANDLER_TRAP(func) void func() __trap

  handler for trap service routine
- #define NOP() __asm__("nop")

  perform a nop() operation (=minimum delay)
- #define DISABLE_INTERRUPTS() __asm__("sim")

  disable interrupt handling
- #define ENABLE_INTERRUPTS() __asm__("rim")

  enable interrupt handling
- #define TRIGGER_TRAP __asm__("trap")

  trigger a trap (=soft interrupt) e.g. for EMC robustness (see AN1015)
- #define WAIT_FOR_INTERRUPT() __asm__("wfi")

  stop code execution and wait for interrupt
- #define ENTER_HALT() __asm__("halt")

  put controller to HALT mode
- #define SW_RESET() (_WWDG_CR=0xBF)

  reset controller via WWGD module
- #define _BITS unsigned int

  data type in bit structs (follow C90 standard)
- #define _SFR(type, addr) (*((volatile type*) (addr)))

  peripheral register
- #define __FLASH_VECTOR__ 1

  irq1 - flash interrupt
- #define __AWU_VECTOR__ 4

  irq4 - Auto Wake Up from Halt interrupt (AWU)
- #define __PORTB_VECTOR__ 6

  irq6 - External interrupt port B
- #define __PORTD_VECTOR__ 7

  irq7 - External interrupt port D
- #define __EXTI0_VECTOR__ 8

  irq8 - External interrupt 0
- #define __EXTI1_VECTOR__ 9

  irq9 - External interrupt 1
- #define __EXTI2_VECTOR__ 10

  irq10 - External interrupt 2
- #define __EXTI3_VECTOR__ 11

  irq11 - External interrupt 3
- #define __EXTI4_VECTOR__ 12

  irq12 - External interrupt 4
- #define __EXTI5_VECTOR__ 13

  irq13 - External interrupt 5
- #define __EXTI6_VECTOR__ 14

  irq14 - External interrupt 6
- #define __EXTI7_VECTOR__ 15

  irq15 - External interrupt 7
- #define __COMP_VECTOR__ 18

  irq18 - comparator interrupt
- #define __TIM2_UPD_OVF_VECTOR__ 19

  irq19 - TIM2 Update/overflow/trigger/break interrupt
- #define __TIM2_CAPCOM_VECTOR__ 20

irq20 - TIM2 Capture/Compare interrupt

- #define __TIM3_UPD_OVF_VECTOR__ 21

  irq21 - TIM3 Update/overflow/break interrupt
- #define __TIM3_CAPCOM_VECTOR__ 22

  irq22 - TIM3 Capture/Compare interrupt
- #define __TIM4_UPD_VECTOR__ 25

  irq25 - TIM4 Update/trigger interrupt
- #define __SPI_VECTOR__ 26

  irq26 - SPI End of transfer interrupt
- #define __USART_TXE_VECTOR__ 27

  irq27 - USART send (TX empty) interrupt
- #define __USART_RXF_VECTOR__ 28

  irq28 - USART receive (RX full) interrupt
- #define __I2C_VECTOR__ 19

  irq29 - I2C interrupt
- #define _GPIOA _SFR(PORT_t, PORTA_AddressBase)

  port A struct/bit access
- #define _GPIOA_ODR _SFR(uint8_t, PORTA_AddressBase+0x00)

  port A output register
- #define _GPIOA_IDR _SFR(uint8_t, PORTA_AddressBase+0x01)

  port A input register
- #define _GPIOA_DDR _SFR(uint8_t, PORTA_AddressBase+0x02)

  port A direction register
- #define _GPIOA_CR1 _SFR(uint8_t, PORTA_AddressBase+0x03)

  port A control register 1
- #define _GPIOA_CR2 _SFR(uint8_t, PORTA_AddressBase+0x04)

  port A control register 2
- #define _GPIOB _SFR(PORT_t, PORTB_AddressBase)

  port B struct/bit access
- #define _GPIOB_ODR _SFR(uint8_t, PORTB_AddressBase+0x00)

  port B output register
- #define _GPIOB_IDR _SFR(uint8_t, PORTB_AddressBase+0x01)

  port B input register
- #define _GPIOB_DDR _SFR(uint8_t, PORTB_AddressBase+0x02)

  port B direction register
- #define _GPIOB_CR1 _SFR(uint8_t, PORTB_AddressBase+0x03)

  port B control register 1
- #define _GPIOB_CR2 _SFR(uint8_t, PORTB_AddressBase+0x04)

  port B control register 2
- #define _GPIOC _SFR(PORT_t, PORTC_AddressBase)

  port C struct/bit access
- #define _GPIOC_ODR _SFR(uint8_t, PORTC_AddressBase+0x00)

  port C output register
- #define _GPIOC_IDR _SFR(uint8_t, PORTC_AddressBase+0x01)

  port C input register
- #define _GPIOC_DDR _SFR(uint8_t, PORTC_AddressBase+0x02)

  port C direction register
- #define _GPIOC_CR1 _SFR(uint8_t, PORTC_AddressBase+0x03)

  port C control register 1
- #define _GPIOC_CR2 _SFR(uint8_t, PORTC_AddressBase+0x04)

  port C control register 2

- #define _GPIOD _SFR(PORT_t, PORTD_AddressBase)
  port D struct/bit access
- #define _GPIOD_ODR _SFR(uint8_t, PORTD_AddressBase+0x00)
  port D output register
- #define _GPIOD_IDR _SFR(uint8_t, PORTD_AddressBase+0x01)
  port D input register
- #define _GPIOD_DDR _SFR(uint8_t, PORTD_AddressBase+0x02)
  port D direction register
- #define _GPIOD_CR1 _SFR(uint8_t, PORTD_AddressBase+0x03)
  port D control register 1
- #define _GPIOD_CR2 _SFR(uint8_t, PORTD_AddressBase+0x04)
  port D control register 2
- #define _GPIO_ODR_RESET_VALUE ((uint8_t) 0x00)
  port output register reset value
- #define _GPIO_DDR_RESET_VALUE ((uint8_t) 0x00)
  port direction register reset value
- #define _GPIO_CR1_RESET_VALUE ((uint8_t) 0x00)
  port control register 1 reset value
- #define _GPIO_CR2_RESET_VALUE ((uint8_t) 0x00)
  port control register 2 reset value
- #define _GPIO_PIN0 ((uint8_t) (0x01 << 0))
  port bit mask for pin 0 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN1 ((uint8_t) (0x01 << 1))
  port bit mask for pin 1 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN2 ((uint8_t) (0x01 << 2))
  port bit mask for pin 2 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN3 ((uint8_t) (0x01 << 3))
  port bit mask for pin 3 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN4 ((uint8_t) (0x01 << 4))
  port bit mask for pin 4 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN5 ((uint8_t) (0x01 << 5))
  port bit mask for pin 5 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN6 ((uint8_t) (0x01 << 6))
  port bit mask for pin 6 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _GPIO_PIN7 ((uint8_t) (0x01 << 7))
  port bit mask for pin 7 (in _GPIOI_ODR, _GPIOI_IDR, _GPIOI_DDR, _GPIOI_CR1, _GPIOI_CR2)
- #define _FLASH _SFR(FLASH_t, FLASH_AddressBase)
  Flash struct/bit access.
- #define _FLASH_CR1 _SFR(uint8_t, FLASH_AddressBase+0x00)
  Flash control register 1 (FLASH_CR1)
- #define _FLASH_CR2 _SFR(uint8_t, FLASH_AddressBase+0x01)
  Flash control register 2 (FLASH_CR2)
- #define _FLASH_PUKR _SFR(uint8_t, FLASH_AddressBase+0x02)
  Flash program memory unprotecting key register (FLASH_PUKR)
- #define _FLASH_DUKR _SFR(uint8_t, FLASH_AddressBase+0x03)
  Data EEPROM unprotection key register (FLASH_DUKR)
- #define _FLASH_IAPSR _SFR(uint8_t, FLASH_AddressBase+0x04)
  Flash status register (FLASH_IAPSR)
- #define _FLASH_CR1_RESET_VALUE ((uint8_t) 0x00)
  Flash control register 1 reset value.
- #define _FLASH_CR2_RESET_VALUE ((uint8_t) 0x00)

Flash control register 2 reset value.

- #define _FLASH_PUKR_RESET_VALUE ((uint8_t) 0x00)

  Flash program memory unprotecting key reset value.
- #define _FLASH_DUKR_RESET_VALUE ((uint8_t) 0x00)

  Data EEPROM unprotection key reset value.
- #define _FLASH_IAPSR_RESET_VALUE ((uint8_t) 0x40)

  Flash status register reset value.
- #define _FLASH_FIX ((uint8_t) (0x01 << 0))

  Fixed Byte programming time [0] (in _FLASH_CR1)
- #define _FLASH_IE ((uint8_t) (0x01 << 1))

  Flash Interrupt enable [0] (in _FLASH_CR1)
- #define _FLASH_PRG ((uint8_t) (0x01 << 0))

  Standard block programming [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_FPRG ((uint8_t) (0x01 << 4))

  Fast block programming [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_ERASE ((uint8_t) (0x01 << 5))

  Block erasing [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_WPRG ((uint8_t) (0x01 << 6))

  Word programming [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_OPT ((uint8_t) (0x01 << 7))

  Write option bytes [0] (in _FLASH_CR2 and _FLASH_NCR2)
- #define _FLASH_WR_PG_DIS ((uint8_t) (0x01 << 0))

  Write attempted to protected page flag [0] (in _FLASH_IAPSR)
- #define _FLASH_PUL ((uint8_t) (0x01 << 1))

  Flash Program memory unlocked flag [0] (in _FLASH_IAPSR)
- #define _FLASH_EOP ((uint8_t) (0x01 << 2))

  End of programming (write or erase operation) flag [0] (in _FLASH_IAPSR)
- #define _FLASH_DUL ((uint8_t) (0x01 << 3))

  Data EEPROM area unlocked flag [0] (in _FLASH_IAPSR)
- #define _EXTI _SFR(EXTI_t, EXTI_AddressBase)

  External interrupt struct/bit access.
- #define _EXTI_CR1 _SFR(uint8_t, EXTI_AddressBase+0x00)

  External interrupt control register 1 (EXTI_CR1)
- #define _EXTI_CR2 _SFR(uint8_t, EXTI_AddressBase+0x01)

  External interrupt control register 2 (EXTI_CR2)
- #define _EXTI_CR3 _SFR(uint8_t, EXTI_AddressBase+0x02)

  External interrupt control register 3 (EXTI_CR2)
- #define _EXTI_SR1 _SFR(uint8_t, EXTI_AddressBase+0x03)

  External interrupt status register 1 (EXTI_SR1)
- #define _EXTI_SR2 _SFR(uint8_t, EXTI_AddressBase+0x04)

  External interrupt status register 2 (EXTI_SR2)
- #define _EXTI_CONF _SFR(uint8_t, EXTI_AddressBase+0x05)

  External interrupt port selector (EXTI_CONF)
- #define _EXTI_CR1_RESET_VALUE ((uint8_t) 0x00)

  External interrupt control register 1 reset value.
- #define _EXTI_CR2_RESET_VALUE ((uint8_t) 0x00)

  External interrupt control register 2 reset value.
- #define _EXTI_CR3_RESET_VALUE ((uint8_t) 0x00)

  External interrupt control register 3 reset value.
- #define _EXTI_SR1_RESET_VALUE ((uint8_t) 0x00)

  External interrupt status register 1 reset value.

- #define _EXTI_SR2_RESET_VALUE ((uint8_t) 0x00)

  External interrupt status register 2 reset value.
- #define _EXTI_CONF_RESET_VALUE ((uint8_t) 0x00)

  External interrupt port selector reset value.
- #define _EXTI_P0IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 0 [1:0] (in _EXTI_CR1)
- #define _EXTI_P0IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 0 [0] (in _EXTI_CR1)
- #define _EXTI_P0IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 0 [1] (in _EXTI_CR1)
- #define _EXTI_P1IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 1 [1:0] (in _EXTI_CR1)
- #define _EXTI_P1IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 1 [0] (in _EXTI_CR1)
- #define _EXTI_P1IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 1 [1] (in _EXTI_CR1)
- #define _EXTI_P2IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 2 [1:0] (in _EXTI_CR1)
- #define _EXTI_P2IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 2 [0] (in _EXTI_CR1)
- #define _EXTI_P2IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 2 [1] (in _EXTI_CR1)
- #define _EXTI_P3IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 3 [1:0] (in _EXTI_CR1)
- #define _EXTI_P3IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 3 [0] (in _EXTI_CR1)
- #define _EXTI_P3IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 3 [1] (in _EXTI_CR1)
- #define _EXTI_P4IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 4 [1:0] (in _EXTI_CR2)
- #define _EXTI_P4IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 4 [0] (in _EXTI_CR2)
- #define _EXTI_P4IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 4 [1] (in _EXTI_CR2)
- #define _EXTI_P5IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 5 [1:0] (in _EXTI_CR2)
- #define _EXTI_P5IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 5 [0] (in _EXTI_CR2)
- #define _EXTI_P5IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 5 [1] (in _EXTI_CR2)
- #define _EXTI_P6IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 6 [1:0] (in _EXTI_CR2)
- #define _EXTI_P6IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 6 [0] (in _EXTI_CR2)
- #define _EXTI_P6IS1 ((uint8_t) (0x01 << 1))

  External interrupt sensitivity for Portx bit 6 [1] (in _EXTI_CR2)
- #define _EXTI_P7IS ((uint8_t) (0x03 << 0))

  External interrupt sensitivity for Portx bit 7 [1:0] (in _EXTI_CR2)
- #define _EXTI_P7IS0 ((uint8_t) (0x01 << 0))

  External interrupt sensitivity for Portx bit 7 [0] (in _EXTI_CR2)
- #define _EXTI_P7IS1 ((uint8_t) (0x01 << 1))

External interrupt sensitivity for Portx bit 7 [1] (in _EXTI_CR2)

- #define _EXTI_PBIS ((uint8_t) (0x03 << 0))

  Port B external interrupt sensitivity bits [1:0] (in _EXTI_CR3)
- #define _EXTI_PBIS0 ((uint8_t) (0x01 << 0))

  Port B external interrupt sensitivity bits [0] (in _EXTI_CR3)
- #define _EXTI_PBIS1 ((uint8_t) (0x01 << 1))

  Port B external interrupt sensitivity bits [1] (in _EXTI_CR3)
- #define _EXTI_PDIS ((uint8_t) (0x03 << 0))

  Port D external interrupt sensitivity bits [1:0] (in _EXTI_CR3)
- #define _EXTI_PDIS0 ((uint8_t) (0x01 << 0))

  Port D external interrupt sensitivity bits [0] (in _EXTI_CR3)
- #define _EXTI_PDIS1 ((uint8_t) (0x01 << 1))

  Port D external interrupt sensitivity bits [1] (in _EXTI_CR3)
- #define _EXTI_P0F ((uint8_t) (0x01 << 0))

  Portx bit 0 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P1F ((uint8_t) (0x02 << 0))

  Portx bit 1 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P2F ((uint8_t) (0x04 << 0))

  Portx bit 2 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P3F ((uint8_t) (0x08 << 0))

  Portx bit 3 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P4F ((uint8_t) (0x10 << 0))

  Portx bit 4 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P5F ((uint8_t) (0x20 << 0))

  Portx bit 5 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P6F ((uint8_t) (0x40 << 0))

  Portx bit 6 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_P7F ((uint8_t) (0x80 << 0))

  Portx bit 7 external interrupt flag (in _EXTI_SR1)
- #define _EXTI_PBF ((uint8_t) (0x01 << 0))

  Port B external interrupt flag (in _EXTI_SR2)
- #define _EXTI_PDF ((uint8_t) (0x02 << 0))

  Port D external interrupt flag (in _EXTI_SR2)
- #define _EXTI_PBLIS ((uint8_t) (0x01 << 0))

  Port B, pins 0..3 external interrupt select (in _EXTI_CONF)
- #define _EXTI_PBHIS ((uint8_t) (0x02 << 0))

  Port B, pins 4..7 external interrupt select (in _EXTI_CONF)
- #define _EXTI_PDLIS ((uint8_t) (0x01 << 0))

  Port D, pins 0..3 external interrupt select (in _EXTI_CONF)
- #define _EXTI_PDHIS ((uint8_t) (0x02 << 0))

  Port D, pins 4..7 external interrupt select (in _EXTI_CONF)
- #define _RST _SFR(RST_t, RST_AddressBase)

  Reset module struct/bit access.
- #define _RST_CR _SFR(uint8_t, RST_AddressBase+0x00)

  Reset pin configuration register (RST_CR)
- #define _RST_SR _SFR(uint8_t, RST_AddressBase+0x01)

  Reset module status register (RST_SR)
- #define _RST_PIN_KEY ((uint8_t) 0xD0)

  Configure PA1 as GPIO, else NRST (in _RST_CR)
- #define _RST_PORF ((uint8_t) (0x01 << 0))

  Power-on reset (POR) flag [0] (in _RST_SR)

- #define _RST_IWDGF ((uint8_t) (0x01 << 1))
  Independent Watchdog reset flag [0] (in _RST_SR)
- #define _RST_ILLOPF ((uint8_t) (0x01 << 2))
  Illegal opcode reset flag [0] (in _RST_SR)
- #define _RST_SWIMF ((uint8_t) (0x01 << 3))
  SWIM reset flag [0] (in _RST_SR)
- #define _CLK _SFR(CLK_t, CLK_AddressBase)
  Clock module struct/bit access.
- #define _CLK_CKDIVR _SFR(uint8_t, CLK_AddressBase+0x00)
  Clock Divider Register.
- #define _CLK_PCKENR _SFR(uint8_t, CLK_AddressBase+0x03)
  Peripheral clock gating register.
- #define _CLK_CCOR _SFR(uint8_t, CLK_AddressBase+0x05)
  Configurable clock output register.
- #define _CLK_CKDIVR_RESET_VALUE ((uint8_t) 0x03)
  Clock divider register reset value.
- #define _CLK_PCKENR_RESET_VALUE ((uint8_t) 0x00)
  Peripheral clock gating register reset value.
- #define _CLK_CCOR_RESET_VALUE ((uint8_t) 0x00)
  Configurable clock output register reset value.
- #define _CLK_HSIDIV ((uint8_t) (0x03 << 0))
  High speed internal clock prescaler [1:0] (in _CLK_CKDIVR)
- #define _CLK_HSIDIV0 ((uint8_t) (0x01 << 0))
  High speed internal clock prescaler [0] (in _CLK_CKDIVR)
- #define _CLK_HSIDIV1 ((uint8_t) (0x01 << 1))
  High speed internal clock prescaler [1] (in _CLK_CKDIVR)
- #define _CLK_HSIDIV_DIV1 ((uint8_t) (0x00 << 0))
  set HSI prescaler to 1 (in _CLK_CKDIVR)
- #define _CLK_HSIDIV_DIV2 ((uint8_t) (0x01 << 0))
  set HSI prescaler to 1/2 (in _CLK_CKDIVR)
- #define _CLK_HSIDIV_DIV4 ((uint8_t) (0x02 << 0))
  set HSI prescaler to 1/4 (in _CLK_CKDIVR)
- #define _CLK_HSIDIV_DIV8 ((uint8_t) (0x03 << 0))
  set HSI prescaler to 1/8 (in _CLK_CKDIVR)
- #define _CLK_TIM2 ((uint8_t) (0x01 << 0))
  clock enable TIM2 [0] (in _CLK_PCKENR1)
- #define _CLK_TIM3 ((uint8_t) (0x01 << 1))
  clock enable TIM3 [0] (in _CLK_PCKENR1)
- #define _CLK_TIM4 ((uint8_t) (0x01 << 2))
  clock enable TIM4 [0] (in _CLK_PCKENR1)
- #define _CLK_I2C ((uint8_t) (0x01 << 3))
  clock enable I2C [0] (in _CLK_PCKENR1)
- #define _CLK_SPI ((uint8_t) (0x01 << 4))
  clock enable SPI [0] (in _CLK_PCKENR1)
- #define _CLK_USART ((uint8_t) (0x01 << 5))
  clock enable USART [0] (in _CLK_PCKENR1)
- #define _CLK_AWU_BEEP ((uint8_t) (0x01 << 6))
  clock enable AWU/BEEP [0] (in _CLK_PCKENR1)
- #define _CLK_CCOEN ((uint8_t) (0x01 << 0))
  Configurable clock output enable [0] (in _CLK_CCOR)
- #define _CLK_CCOSEL ((uint8_t) (0x03 << 1))

Configurable clock output selection [1:0] (in _CLK_CCOR)

- #define _CLK_CCOSEL0 ((uint8_t) (0x01 << 1))

  Configurable clock output selection [0] (in _CLK_CCOR)
- #define _CLK_CCOSEL1 ((uint8_t) (0x01 << 2))

  Configurable clock output selection [1] (in _CLK_CCOR)
- #define _CLK_CCOSEL_DIV1 ((uint8_t) (0x00 << 1))

  set clock output selection to 1 (in _CLK_CCOR)
- #define _CLK_CCOSEL_DIV2 ((uint8_t) (0x01 << 1))

  set clock output selection to 1/2 (in _CLK_CCOR)
- #define _CLK_CCOSEL_DIV4 ((uint8_t) (0x02 << 1))

  set clock output selection to 1/4 (in _CLK_CCOR)
- #define _CLK_CCOSEL_DIV16 ((uint8_t) (0x03 << 1))

  set clock output selection to 1/16 (in _CLK_CCOR)
- #define _IWDG _SFR(IWDG_t, IWDG_AddressBase)

  Independent Timeout Watchdog struct/bit access.
- #define _IWDG_KR _SFR(uint8_t, IWDG_AddressBase+0x00)

  Independent Timeout Watchdog Key register (IWDG_KR)
- #define _IWDG_PR _SFR(uint8_t, IWDG_AddressBase+0x01)

  Independent Timeout Watchdog Prescaler register (IWDG_PR)
- #define _IWDG_RLR _SFR(uint8_t, IWDG_AddressBase+0x02)

  Independent Timeout Watchdog Reload register (IWDG_RLR)
- #define _IWDG_PR_RESET_VALUE ((uint8_t) 0x00)

  Independent Timeout Watchdog Prescaler register reset value.
- #define _IWDG_RLR_RESET_VALUE ((uint8_t) 0xFF)

  Independent Timeout Watchdog Reload register reset value.
- #define _IWDG_KEY_ENABLE ((uint8_t) 0xCC)

  Independent Timeout Watchdog enable (in _IWDG_KR)
- #define _IWDG_KEY_REFRESH ((uint8_t) 0xAA)

  Independent Timeout Watchdog refresh (in _IWDG_KR)
- #define _IWDG_KEY_ACCESS ((uint8_t) 0x55)

  Independent Timeout Watchdog unlock write to IWDG_PR and IWDG_RLR (in _IWDG_KR)
- #define _IWDG_PRE ((uint8_t) (0x07 << 0))

  Independent Timeout Watchdog Prescaler divider [2:0] (in _IWDG_PR)
- #define _IWDG_PRE0 ((uint8_t) (0x01 << 0))

  Independent Timeout Watchdog Prescaler divider [0] (in _IWDG_PR)
- #define _IWDG_PRE1 ((uint8_t) (0x01 << 1))

  Independent Timeout Watchdog Prescaler divider [1] (in _IWDG_PR)
- #define _IWDG_PRE2 ((uint8_t) (0x01 << 2))

  Independent Timeout Watchdog Prescaler divider [2] (in _IWDG_PR)
- #define _AWU _SFR(AWU_t, AWU_AddressBase)

  Auto Wake-Up struct/bit access.
- #define _AWU_CSR _SFR(uint8_t, AWU_AddressBase+0x00)

  Auto Wake-Up Control/status register (AWU_CSR)
- #define _AWU_APR _SFR(uint8_t, AWU_AddressBase+0x01)

  Auto Wake-Up Asynchronous prescaler register (AWU_APR)
- #define _AWU_TBR _SFR(uint8_t, AWU_AddressBase+0x02)

  Auto Wake-Up Timebase selection register (AWU_TBR)
- #define _AWU_CSR_RESET_VALUE ((uint8_t) 0x00)

  Auto Wake-Up Control/status register reset value.
- #define _AWU_APR_RESET_VALUE ((uint8_t) 0x3F)

  Auto Wake-Up Asynchronous prescaler register reset value.

- #define _AWU_TBR_RESET_VALUE ((uint8_t) 0x00)

  Auto Wake-Up Timebase selection register reset value.
- #define _AWU_MSR ((uint8_t) (0x01 << 0))

  Auto Wake-Up LSI measurement enable [0] (in _AWU_CSR)
- #define _AWU_AWUEN ((uint8_t) (0x01 << 4))

  Auto-wakeup enable [0] (in _AWU_CSR)
- #define _AWU_AWUF ((uint8_t) (0x01 << 5))

  Auto-wakeup status flag [0] (in _AWU_CSR)
- #define _AWU_APRE ((uint8_t) (0x3F << 0))

  Auto-wakeup asynchronous prescaler divider [5:0] (in _AWU_APR)
- #define _AWU_APRE0 ((uint8_t) (0x01 << 0))

  Auto-wakeup asynchronous prescaler divider [0] (in _AWU_APR)
- #define _AWU_APRE1 ((uint8_t) (0x01 << 1))

  Auto-wakeup asynchronous prescaler divider [1] (in _AWU_APR)
- #define _AWU_APRE2 ((uint8_t) (0x01 << 2))

  Auto-wakeup asynchronous prescaler divider [2] (in _AWU_APR)
- #define _AWU_APRE3 ((uint8_t) (0x01 << 3))

  Auto-wakeup asynchronous prescaler divider [3] (in _AWU_APR)
- #define _AWU_APRE4 ((uint8_t) (0x01 << 4))

  Auto-wakeup asynchronous prescaler divider [4] (in _AWU_APR)
- #define _AWU_APRE5 ((uint8_t) (0x01 << 5))

  Auto-wakeup asynchronous prescaler divider [5] (in _AWU_APR)
- #define _AWU_AWUTB ((uint8_t) (0x0F << 0))

  Auto-wakeup timebase selection [3:0] (in _AWU_APR)
- #define _AWU_AWUTB0 ((uint8_t) (0x01 << 0))

  Auto-wakeup timebase selection [0] (in _AWU_APR)
- #define _AWU_AWUTB1 ((uint8_t) (0x01 << 1))

  Auto-wakeup timebase selection [1] (in _AWU_APR)
- #define _AWU_AWUTB2 ((uint8_t) (0x01 << 2))

  Auto-wakeup timebase selection [2] (in _AWU_APR)
- #define _AWU_AWUTB3 ((uint8_t) (0x01 << 3))

  Auto-wakeup timebase selection [3] (in _AWU_APR)
- #define _BEEP _SFR(BEEP_t, BEEP_AddressBase)

  Beeper struct/bit access.
- #define _BEEP_CSR _SFR(uint8_t, BEEP_AddressBase+0x00)

  Beeper control/status register (BEEP_CSR)
- #define _BEEP_CSR_RESET_VALUE ((uint8_t) 0x1F)

  Beeper control/status register reset value.
- #define _BEEP_BEEPDIV ((uint8_t) (0x1F << 0))

  Beeper clock prescaler divider [4:0] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV0 ((uint8_t) (0x01 << 0))

  Beeper clock prescaler divider [0] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV1 ((uint8_t) (0x01 << 1))

  Beeper clock prescaler divider [1] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV2 ((uint8_t) (0x01 << 2))

  Beeper clock prescaler divider [2] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV3 ((uint8_t) (0x01 << 3))

  Beeper clock prescaler divider [3] (in _BEEP_CSR)
- #define _BEEP_BEEPDIV4 ((uint8_t) (0x01 << 4))

  Beeper clock prescaler divider [4] (in _BEEP_CSR)
- #define _BEEP_BEEPEN ((uint8_t) (0x01 << 5))

Beeper enable [0] (in _BEEP_CSR)

- #define _BEEP_BEEPSEL ((uint8_t) (0x03 << 6))

  Beeper frequency selection [1:0] (in _BEEP_CSR)
- #define _BEEP_BEEPSEL0 ((uint8_t) (0x01 << 6))

  Beeper frequency selection [0] (in _BEEP_CSR)
- #define _BEEP_BEEPSEL1 ((uint8_t) (0x01 << 7))

  Beeper frequency selection [1] (in _BEEP_CSR)
- #define _SPI _SFR(SPI_t, SPI_AddressBase)

  register for SPI control
- #define _SPI_CR1 _SFR(uint8_t, SPI_AddressBase+0x00)

  SPI control register 1.
- #define _SPI_CR2 _SFR(uint8_t, SPI_AddressBase+0x01)

  SPI control register 2.
- #define _SPI_ICR _SFR(uint8_t, SPI_AddressBase+0x02)

  SPI interrupt control register.
- #define _SPI_SR _SFR(uint8_t, SPI_AddressBase+0x03)

  SPI status register.
- #define _SPI_DR _SFR(uint8_t, SPI_AddressBase+0x04)

  SPI data register.
- #define _SPI_CR1_RESET_VALUE ((uint8_t) 0x00)

  SPI Control Register 1 reset value.
- #define _SPI_CR2_RESET_VALUE ((uint8_t) 0x00)

  SPI Control Register 2 reset value.
- #define _SPI_ICR_RESET_VALUE ((uint8_t) 0x00)

  SPI Interrupt Control Register reset value.
- #define _SPI_SR_RESET_VALUE ((uint8_t) 0x02)

  SPI Status Register reset value.
- #define _SPI_DR_RESET_VALUE ((uint8_t) 0x00)

  SPI Data Register reset value.
- #define _SPI_CPHA ((uint8_t) (0x01 << 0))

  SPI Clock phase [0] (in _SPI_CR1)
- #define _SPI_CPOL ((uint8_t) (0x01 << 1))

  SPI Clock polarity [0] (in _SPI_CR1)
- #define _SPI_MSTR ((uint8_t) (0x01 << 2))

  SPI Master/slave selection [0] (in _SPI_CR1)
- #define _SPI_BR ((uint8_t) (0x07 << 3))

  SPI Baudrate control [2:0] (in _SPI_CR1)
- #define _SPI_BR0 ((uint8_t) (0x01 << 3))

  SPI Baudrate control [0] (in _SPI_CR1)
- #define _SPI_BR1 ((uint8_t) (0x01 << 4))

  SPI Baudrate control [1] (in _SPI_CR1)
- #define _SPI_BR2 ((uint8_t) (0x01 << 5))

  SPI Baudrate control [2] (in _SPI_CR1)
- #define _SPI_SPE ((uint8_t) (0x01 << 6))

  SPI enable [0] (in _SPI_CR1)
- #define _SPI_LSBFIRST ((uint8_t) (0x01 << 7))

  SPI Frame format [0] (in _SPI_CR1)
- #define _SPI_SSI ((uint8_t) (0x01 << 0))

  SPI Internal slave select [0] (in _SPI_CR2)
- #define _SPI_SSM ((uint8_t) (0x01 << 1))

  SPI Software slave management [0] (in _SPI_CR2)

- #define _SPI_RXONLY ((uint8_t) (0x01 << 2))
  SPI Receive only [0] (in _SPI_CR2)
- #define _SPI_BDOE ((uint8_t) (0x01 << 6))
  SPI Input/Output enable in bidirectional mode [0] (in _SPI_CR2)
- #define _SPI_BDM ((uint8_t) (0x01 << 7))
  SPI Bidirectional data mode enable [0] (in _SPI_CR2)
- #define _SPI_WKIE ((uint8_t) (0x01 << 4))
  SPI Wakeup interrupt enable [0] (in _SPI_ICR)
- #define _SPI_ERRIE ((uint8_t) (0x01 << 5))
  SPI Error interrupt enable [0] (in _SPI_ICR)
- #define _SPI_RXIE ((uint8_t) (0x01 << 6))
  SPI Rx buffer not empty interrupt enable [0] (in _SPI_ICR)
- #define _SPI_TXIE ((uint8_t) (0x01 << 7))
  SPI Tx buffer empty interrupt enable [0] (in _SPI_ICR)
- #define _SPI_RXNE ((uint8_t) (0x01 << 0))
  SPI Receive buffer not empty [0] (in _SPI_SR)
- #define _SPI_TXE ((uint8_t) (0x01 << 1))
  SPI Transmit buffer empty [0] (in _SPI_SR)
- #define _SPI_WKUP ((uint8_t) (0x01 << 3))
  SPI Wakeup flag [0] (in _SPI_SR)
- #define _SPI_MODF ((uint8_t) (0x01 << 5))
  SPI Mode fault [0] (in _SPI_SR)
- #define _SPI_OVR ((uint8_t) (0x01 << 6))
  SPI Overrun flag [0] (in _SPI_SR)
- #define _SPI_BSY ((uint8_t) (0x01 << 7))
  SPI Busy flag [0] (in _SPI_SR)
- #define _I2C _SFR(I2C_t, I2C_AddressBase)
  register for SPI control
- #define _I2C_CR1 _SFR(uint8_t, I2C_AddressBase+0x00)
  I2C Control register 1.
- #define _I2C_CR2 _SFR(uint8_t, I2C_AddressBase+0x01)
  I2C Control register 2.
- #define _I2C_FREQR _SFR(uint8_t, I2C_AddressBase+0x02)
  I2C Frequency register.
- #define _I2C_OARL _SFR(uint8_t, I2C_AddressBase+0x03)
  I2C own address register low byte.
- #define _I2C_OARH _SFR(uint8_t, I2C_AddressBase+0x04)
  I2C own address register high byte.
- #define _I2C_DR _SFR(uint8_t, I2C_AddressBase+0x06)
  I2C data register.
- #define _I2C_SR1 _SFR(uint8_t, I2C_AddressBase+0x07)
  I2C Status register 1.
- #define _I2C_SR2 _SFR(uint8_t, I2C_AddressBase+0x08)
  I2C Status register 2.
- #define _I2C_SR3 _SFR(uint8_t, I2C_AddressBase+0x09)
  I2C Status register 3.
- #define _I2C_ITR _SFR(uint8_t, I2C_AddressBase+0x0A)
  I2C Interrupt register.
- #define _I2C_CCRL _SFR(uint8_t, I2C_AddressBase+0x0B)
  I2C Clock control register low byte.
- #define _I2C_CCRH _SFR(uint8_t, I2C_AddressBase+0x0C)

I2C Clock control register high byte.

- #define _I2C_TRISER _SFR(uint8_t, I2C_AddressBase+0x0D)

  I2C rise time register.
- #define _I2C_CR1_RESET_VALUE ((uint8_t) 0x00)

  I2C Control register 1 reset value.
- #define _I2C_CR2_RESET_VALUE ((uint8_t) 0x00)

  I2C Control register 2 reset value.
- #define _I2C_FREQR_RESET_VALUE ((uint8_t) 0x00)

  I2C Frequency register reset value.
- #define _I2C_OARL_RESET_VALUE ((uint8_t) 0x00)

  I2C own address register low byte reset value.
- #define _I2C_OARH_RESET_VALUE ((uint8_t) 0x00)

  I2C own address register high byte reset value.
- #define _I2C_DR_RESET_VALUE ((uint8_t) 0x00)

  I2C data register reset value.
- #define _I2C_SR1_RESET_VALUE ((uint8_t) 0x00)

  I2C Status register 1 reset value.
- #define _I2C_SR2_RESET_VALUE ((uint8_t) 0x00)

  I2C Status register 2 reset value.
- #define _I2C_SR3_RESET_VALUE ((uint8_t) 0x00)

  I2C Status register 3 reset value.
- #define _I2C_ITR_RESET_VALUE ((uint8_t) 0x00)

  I2C Interrupt register reset value.
- #define _I2C_CCRL_RESET_VALUE ((uint8_t) 0x00)

  I2C Clock control register low byte reset value.
- #define _I2C_CCRH_RESET_VALUE ((uint8_t) 0x00)

  I2C Clock control register high byte reset value.
- #define _I2C_TRISER_RESET_VALUE ((uint8_t) 0x02)

  I2C rise time register reset value.
- #define _I2C_PE ((uint8_t) (0x01 << 0))

  I2C Peripheral enable [0] (in _I2C_CR1)
- #define _I2C_ENGC ((uint8_t) (0x01 << 6))

  I2C General call enable [0] (in _I2C_CR1)
- #define _I2C_NOSTRETCH ((uint8_t) (0x01 << 7))

  I2C Clock stretching disable (Slave mode) [0] (in _I2C_CR1)
- #define _I2C_START ((uint8_t) (0x01 << 0))

  I2C Start generation [0] (in _I2C_CR2)
- #define _I2C_STOP ((uint8_t) (0x01 << 1))

  I2C Stop generation [0] (in _I2C_CR2)
- #define _I2C_ACK ((uint8_t) (0x01 << 2))

  I2C Acknowledge enable [0] (in _I2C_CR2)
- #define _I2C_POS ((uint8_t) (0x01 << 3))

  I2C Acknowledge position (for data reception) [0] (in _I2C_CR2)
- #define _I2C_SWRST ((uint8_t) (0x01 << 7))

  I2C Software reset [0] (in _I2C_CR2)
- #define _I2C_FREQ ((uint8_t) (0x3F << 0))

  I2C Peripheral clock frequency [5:0] (in _I2C_FREQR)
- #define _I2C_FREQ0 ((uint8_t) (0x01 << 0))

  I2C Peripheral clock frequency [0] (in _I2C_FREQR)
- #define _I2C_FREQ1 ((uint8_t) (0x01 << 1))

  I2C Peripheral clock frequency [1] (in _I2C_FREQR)

- #define _I2C_FREQ2 ((uint8_t) (0x01 << 2))

  I2C Peripheral clock frequency [2] (in _I2C_FREQR)
- #define _I2C_FREQ3 ((uint8_t) (0x01 << 3))

  I2C Peripheral clock frequency [3] (in _I2C_FREQR)
- #define _I2C_FREQ4 ((uint8_t) (0x01 << 4))

  I2C Peripheral clock frequency [4] (in _I2C_FREQR)
- #define _I2C_FREQ5 ((uint8_t) (0x01 << 5))

  I2C Peripheral clock frequency [5] (in _I2C_FREQR)
- #define _I2C_ADD0 ((uint8_t) (0x01 << 0))

  I2C Interface address [0] (in 10-bit address mode) (in _I2C_OARL)
- #define _I2C_ADD1 ((uint8_t) (0x01 << 1))

  I2C Interface address [1] (in _I2C_OARL)
- #define _I2C_ADD2 ((uint8_t) (0x01 << 2))

  I2C Interface address [2] (in _I2C_OARL)
- #define _I2C_ADD3 ((uint8_t) (0x01 << 3))

  I2C Interface address [3] (in _I2C_OARL)
- #define _I2C_ADD4 ((uint8_t) (0x01 << 4))

  I2C Interface address [4] (in _I2C_OARL)
- #define _I2C_ADD5 ((uint8_t) (0x01 << 5))

  I2C Interface address [5] (in _I2C_OARL)
- #define _I2C_ADD6 ((uint8_t) (0x01 << 6))

  I2C Interface address [6] (in _I2C_OARL)
- #define _I2C_ADD7 ((uint8_t) (0x01 << 7))

  I2C Interface address [7] (in _I2C_OARL)
- #define _I2C_ADD_8_9 ((uint8_t) (0x03 << 1))

  I2C Interface address [9:8] (in 10-bit address mode) (in _I2C_OARH)
- #define _I2C_ADD8 ((uint8_t) (0x01 << 1))

  I2C Interface address [8] (in _I2C_OARH)
- #define _I2C_ADD9 ((uint8_t) (0x01 << 2))

  I2C Interface address [9] (in _I2C_OARH)
- #define _I2C_ADDCONF ((uint8_t) (0x01 << 6))

  I2C Address mode configuration [0] (in _I2C_OARH)
- #define _I2C_ADDMODE ((uint8_t) (0x01 << 7))

  I2C 7-/10-bit addressing mode (Slave mode) [0] (in _I2C_OARH)
- #define _I2C_SB ((uint8_t) (0x01 << 0))

  I2C Start bit (Master mode) [0] (in _I2C_SR1)
- #define _I2C_ADDR ((uint8_t) (0x01 << 1))

  I2C Address sent (Master mode) / matched (Slave mode) [0] (in _I2C_SR1)
- #define _I2C_BTF ((uint8_t) (0x01 << 2))

  I2C Byte transfer finished [0] (in _I2C_SR1)
- #define _I2C_ADD10 ((uint8_t) (0x01 << 3))

  I2C 10-bit header sent (Master mode) [0] (in _I2C_SR1)
- #define _I2C_STOPF ((uint8_t) (0x01 << 4))

  I2C Stop detection (Slave mode) [0] (in _I2C_SR1)
- #define _I2C_RXNE ((uint8_t) (0x01 << 6))

  I2C Data register not empty (receivers) [0] (in _I2C_SR1)
- #define _I2C_TXE ((uint8_t) (0x01 << 7))

  I2C Data register empty (transmitters) [0] (in _I2C_SR1)
- #define _I2C_BERR ((uint8_t) (0x01 << 0))

  I2C Bus error [0] (in _I2C_SR2)
- #define _I2C_ARLO ((uint8_t) (0x01 << 1))

I2C Arbitration lost (Master mode) [0] (in _I2C_SR2)

- #define _I2C_AF ((uint8_t) (0x01 << 2))

  I2C Acknowledge failure [0] (in _I2C_SR2)
- #define _I2C_OVR ((uint8_t) (0x01 << 3))

  I2C Overrun/underrun [0] (in _I2C_SR2)
- #define _I2C_WUFH ((uint8_t) (0x01 << 5))

  I2C Wakeup from Halt [0] (in _I2C_SR2)
- #define _I2C_MSL ((uint8_t) (0x01 << 0))

  I2C Master/Slave [0] (in _I2C_SR3)
- #define _I2C_BUSY ((uint8_t) (0x01 << 1))

  I2C Bus busy [0] (in _I2C_SR3)
- #define _I2C_TRA ((uint8_t) (0x01 << 2))

  I2C Transmitter/Receiver [0] (in _I2C_SR3)
- #define _I2C_GENCALL ((uint8_t) (0x01 << 4))

  I2C General call header (Slave mode) [0] (in _I2C_SR3)
- #define _I2C_DUALF ((uint8_t) (0x01 << 7))

  Dual flag (Slave mode) [0] (in _I2C_SR3)
- #define _I2C_ITERREN ((uint8_t) (0x01 << 0))

  I2C Error interrupt enable [0] (in _I2C_ITR)
- #define _I2C_ITEVTEN ((uint8_t) (0x01 << 1))

  I2C Event interrupt enable [0] (in _I2C_ITR)
- #define _I2C_ITBUFEN ((uint8_t) (0x01 << 2))

  I2C Buffer interrupt enable [0] (in _I2C_ITR)
- #define _I2C_CCR ((uint8_t) (0x0F << 0))

  I2C Clock control register (Master mode) [3:0] (in _I2C_CCRH)
- #define _I2C_CCR0 ((uint8_t) (0x01 << 0))

  I2C Clock control register (Master mode) [0] (in _I2C_CCRH)
- #define _I2C_CCR1 ((uint8_t) (0x01 << 1))

  I2C Clock control register (Master mode) [1] (in _I2C_CCRH)
- #define _I2C_CCR2 ((uint8_t) (0x01 << 2))

  I2C Clock control register (Master mode) [2] (in _I2C_CCRH)
- #define _I2C_CCR3 ((uint8_t) (0x01 << 3))

  I2C Clock control register (Master mode) [3] (in _I2C_CCRH)
- #define _I2C_DUTY ((uint8_t) (0x01 << 6))

  I2C Fast mode duty cycle [0] (in _I2C_CCRH)
- #define _I2C_FS ((uint8_t) (0x01 << 7))

  I2C Master mode selection [0] (in _I2C_CCRH)
- #define _I2C_TRISE ((uint8_t) (0x3F << 0))

  I2C Maximum rise time (Master mode) [5:0] (in _I2C_TRISER)
- #define _I2C_TRISE0 ((uint8_t) (0x01 << 0))

  I2C Maximum rise time (Master mode) [0] (in _I2C_TRISER)
- #define _I2C_TRISE1 ((uint8_t) (0x01 << 1))

  I2C Maximum rise time (Master mode) [1] (in _I2C_TRISER)
- #define _I2C_TRISE2 ((uint8_t) (0x01 << 2))

  I2C Maximum rise time (Master mode) [2] (in _I2C_TRISER)
- #define _I2C_TRISE3 ((uint8_t) (0x01 << 3))

  I2C Maximum rise time (Master mode) [3] (in _I2C_TRISER)
- #define _I2C_TRISE4 ((uint8_t) (0x01 << 4))

  I2C Maximum rise time (Master mode) [4] (in _I2C_TRISER)
- #define _I2C_TRISE5 ((uint8_t) (0x01 << 5))

  I2C Maximum rise time (Master mode) [5] (in _I2C_TRISER)

- #define _USART _SFR(USART_t, USART_AddressBase)
  USART struct/bit access.
- #define _USART_SR _SFR(uint8_t, USART_AddressBase+0x00)
  USART Status register.
- #define _USART_DR _SFR(uint8_t, USART_AddressBase+0x01)
  USART data register.
- #define _USART_BRR1 _SFR(uint8_t, USART_AddressBase+0x02)
  USART Baud rate register 1.
- #define _USART_BRR2 _SFR(uint8_t, USART_AddressBase+0x03)
  USART Baud rate register 2.
- #define _USART_CR1 _SFR(uint8_t, USART_AddressBase+0x04)
  USART Control register 1.
- #define _USART_CR2 _SFR(uint8_t, USART_AddressBase+0x05)
  USART Control register 2.
- #define _USART_CR3 _SFR(uint8_t, USART_AddressBase+0x06)
  USART Control register 3.
- #define _USART_CR4 _SFR(uint8_t, USART_AddressBase+0x07)
  USART Control register 4.
- #define _USART_SR_RESET_VALUE ((uint8_t) 0xC0)
  USART Status register reset value.
- #define _USART_BRR1_RESET_VALUE ((uint8_t) 0x00)
  USART Baud rate register 1 reset value.
- #define _USART_BRR2_RESET_VALUE ((uint8_t) 0x00)
  USART Baud rate register 2 reset value.
- #define _USART_CR1_RESET_VALUE ((uint8_t) 0x00)
  USART Control register 1 reset value.
- #define _USART_CR2_RESET_VALUE ((uint8_t) 0x00)
  USART Control register 2 reset value.
- #define _USART_CR3_RESET_VALUE ((uint8_t) 0x00)
  USART Control register 3 reset value.
- #define _USART_CR4_RESET_VALUE ((uint8_t) 0x00)
  USART Control register 4 reset value.
- #define _USART_PE ((uint8_t) (0x01 << 0))
  USART Parity error [0] (in _USART_SR)
- #define _USART_FE ((uint8_t) (0x01 << 1))
  USART Framing error [0] (in _USART_SR)
- #define _USART_NF ((uint8_t) (0x01 << 2))
  USART Noise flag [0] (in _USART_SR)
- #define _USART_OR_LHE ((uint8_t) (0x01 << 3))
  USART LIN Header Error (LIN Slave mode) / Overrun error [0] (in _USART_SR)
- #define _USART_IDLE ((uint8_t) (0x01 << 4))
  USART IDLE line detected [0] (in _USART_SR)
- #define _USART_RXNE ((uint8_t) (0x01 << 5))
  USART Read data register not empty [0] (in _USART_SR)
- #define _USART_TC ((uint8_t) (0x01 << 6))
  USART Transmission complete [0] (in _USART_SR)
- #define _USART_TXE ((uint8_t) (0x01 << 7))
  USART Transmit data register empty [0] (in _USART_SR)
- #define _USART_PIEN ((uint8_t) (0x01 << 0))
  USART Parity interrupt enable [0] (in _USART_CR1)
- #define _USART_PS ((uint8_t) (0x01 << 1))

USART Parity selection [0] (in _USART_CR1)

- #define _USART_PCEN ((uint8_t) (0x01 << 2))

  USART Parity control enable [0] (in _USART_CR1)
- #define _USART_WAKE ((uint8_t) (0x01 << 3))

  USART Wakeup method [0] (in _USART_CR1)
- #define _USART_M ((uint8_t) (0x01 << 4))

  USART word length [0] (in _USART_CR1)
- #define _USART_UARTD ((uint8_t) (0x01 << 5))

  USART Disable (for low power consumption) [0] (in _USART_CR1)
- #define _USART_T8 ((uint8_t) (0x01 << 6))

  USART Transmit Data bit 8 (in 9-bit mode) [0] (in _USART_CR1)
- #define _USART_R8 ((uint8_t) (0x01 << 7))

  USART Receive Data bit 8 (in 9-bit mode) [0] (in _USART_CR1)
- #define _USART_SBK ((uint8_t) (0x01 << 0))

  USART Send break [0] (in _USART_CR2)
- #define _USART_RWU ((uint8_t) (0x01 << 1))

  USART Receiver wakeup [0] (in _USART_CR2)
- #define _USART_REN ((uint8_t) (0x01 << 2))

  USART Receiver enable [0] (in _USART_CR2)
- #define _USART_TEN ((uint8_t) (0x01 << 3))

  USART Transmitter enable [0] (in _USART_CR2)
- #define _USART_ILIEN ((uint8_t) (0x01 << 4))

  USART IDLE Line interrupt enable [0] (in _USART_CR2)
- #define _USART_RIEN ((uint8_t) (0x01 << 5))

  USART Receiver interrupt enable [0] (in _USART_CR2)
- #define _USART_TCIEN ((uint8_t) (0x01 << 6))

  USART Transmission complete interrupt enable [0] (in _USART_CR2)
- #define _USART_TIEN ((uint8_t) (0x01 << 7))

  USART Transmitter interrupt enable [0] (in _USART_CR2)
- #define _USART_LBCL ((uint8_t) (0x01 << 0))

  USART Last bit clock pulse [0] (in _USART_CR3)
- #define _USART_CPHA ((uint8_t) (0x01 << 1))

  USART Clock phase [0] (in _USART_CR3)
- #define _USART_CPOL ((uint8_t) (0x01 << 2))

  USART Clock polarity [0] (in _USART_CR3)
- #define _USART_CKEN ((uint8_t) (0x01 << 3))

  USART Clock enable [0] (in _USART_CR3)
- #define _USART_STOP ((uint8_t) (0x03 << 4))

  USART STOP bits [1:0] (in _USART_CR3)
- #define _USART_STOP0 ((uint8_t) (0x01 << 4))

  USART STOP bits [0] (in _USART_CR3)
- #define _USART_STOP1 ((uint8_t) (0x01 << 5))

  USART STOP bits [1] (in _USART_CR3)
- #define _USART_ADD ((uint8_t) (0x0F << 0))

  USART Address of the UART node [3:0] (in _USART_CR4)
- #define _USART_ADD0 ((uint8_t) (0x01 << 0))

  USART Address of the UART node [0] (in _USART_CR4)
- #define _USART_ADD1 ((uint8_t) (0x01 << 1))

  USART Address of the UART node [1] (in _USART_CR4)
- #define _USART_ADD2 ((uint8_t) (0x01 << 2))

  USART Address of the UART node [2] (in _USART_CR4)

- #define _USART_ADD3 ((uint8_t) (0x01 << 3))

  USART Address of the UART node [3] (in _USART_CR4)
- #define _WFE _SFR(WFE_t, WFE_AddressBase)

  WFE struct/bit access.
- #define _WFE_CR1 _SFR(uint8_t, WFE_AddressBase+0x00)

  WFE Control register 1.
- #define _WFE_CR2 _SFR(uint8_t, WFE_AddressBase+0x01)

  WFE Control register 2.
- #define _WFE_CR1_RESET_VALUE ((uint8_t) 0x03)

  WFE Control register 1 reset value.
- #define _WFE_CR2_RESET_VALUE ((uint8_t) 0x00)

  WFE Control register 2 reset value.
- #define _WFE_TIM2_EV0 ((uint8_t) (0x01 << 0))

  TIM2 update, trigger or break event [0] (in _WFE_CR1)
- #define _WFE_TIM2_EV1 ((uint8_t) (0x01 << 1))

  TIM2 capture or compare event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV0 ((uint8_t) (0x01 << 4))

  Interrupt on pin 0 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV1 ((uint8_t) (0x01 << 5))

  Interrupt on pin 1 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV2 ((uint8_t) (0x01 << 6))

  Interrupt on pin 2 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV3 ((uint8_t) (0x01 << 7))

  Interrupt on pin 3 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV4 ((uint8_t) (0x01 << 0))

  Interrupt on pin 4 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV5 ((uint8_t) (0x01 << 1))

  Interrupt on pin 5 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV6 ((uint8_t) (0x01 << 2))

  Interrupt on pin 6 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EV7 ((uint8_t) (0x01 << 3))

  Interrupt on pin 7 of all ports event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EVB ((uint8_t) (0x01 << 4))

  Interrupt on port B event [0] (in _WFE_CR1)
- #define _WFE_EXTI_EVD ((uint8_t) (0x01 << 5))

  Interrupt on port D event [0] (in _WFE_CR1)
- #define _TIM2 _SFR(TIM2_3_t, TIM2_AddressBase)

  TIM2 struct/bit access.
- #define _TIM2_CR1 _SFR(uint8_t, TIM2_AddressBase+0x00)

  TIM2 control register 1.
- #define _TIM2_CR2 _SFR(uint8_t, TIM2_AddressBase+0x01)

  TIM2 control register 2.
- #define _TIM2_SMCR _SFR(uint8_t, TIM2_AddressBase+0x02)

  TIM2 Slave mode control register.
- #define _TIM2_ETR _SFR(uint8_t, TIM2_AddressBase+0x03)

  TIM2 External trigger register.
- #define _TIM2_IER _SFR(uint8_t, TIM2_AddressBase+0x04)

  TIM2 interrupt enable register.
- #define _TIM2_SR1 _SFR(uint8_t, TIM2_AddressBase+0x05)

  TIM2 status register 1.
- #define _TIM2_SR2 _SFR(uint8_t, TIM2_AddressBase+0x06)

TIM2 status register 2.

- #define _TIM2_EGR _SFR(uint8_t, TIM2_AddressBase+0x07)

  TIM2 Event generation register.
- #define _TIM2_CCMR1 _SFR(uint8_t, TIM2_AddressBase+0x08)

  TIM2 Capture/compare mode register 1.
- #define _TIM2_CCMR2 _SFR(uint8_t, TIM2_AddressBase+0x09)

  TIM2 Capture/compare mode register 2.
- #define _TIM2_CCER1 _SFR(uint8_t, TIM2_AddressBase+0x0A)

  TIM2 Capture/compare enable register 1.
- #define _TIM2_CNTRH _SFR(uint8_t, TIM2_AddressBase+0x0B)

  TIM2 counter register high byte.
- #define _TIM2_CNTRL _SFR(uint8_t, TIM2_AddressBase+0x0C)

  TIM2 counter register low byte.
- #define _TIM2_PSCR _SFR(uint8_t, TIM2_AddressBase+0x0D)

  TIM2 clock prescaler register.
- #define _TIM2_ARRH _SFR(uint8_t, TIM2_AddressBase+0x0E)

  TIM2 auto-reload register high byte.
- #define _TIM2_ARRL _SFR(uint8_t, TIM2_AddressBase+0x0F)

  TIM2 auto-reload register low byte.
- #define _TIM2_CCR1H _SFR(uint8_t, TIM2_AddressBase+0x10)

  TIM2 16-bit capture/compare value 1 high byte.
- #define _TIM2_CCR1L _SFR(uint8_t, TIM2_AddressBase+0x11)

  TIM2 16-bit capture/compare value 1 low byte.
- #define _TIM2_CCR2H _SFR(uint8_t, TIM2_AddressBase+0x12)

  TIM2 16-bit capture/compare value 2 high byte.
- #define _TIM2_CCR2L _SFR(uint8_t, TIM2_AddressBase+0x13)

  TIM2 16-bit capture/compare value 2 low byte.
- #define _TIM2_BKR _SFR(uint8_t, TIM2_AddressBase+0x14)

  TIM2 Break register.
- #define _TIM2_OISR _SFR(uint8_t, TIM2_AddressBase+0x15)

  TIM2 Output idle state register.
- #define _TIM2_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 control register 1 reset value.
- #define _TIM2_CR2_RESET_VALUE ((uint8_t) 0x00)

  TIM2 control register 2 reset value.
- #define _TIM2_SMCR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Slave mode control register reset value.
- #define _TIM2_ETR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 External trigger register reset value.
- #define _TIM2_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM2 interrupt enable register reset value.
- #define _TIM2_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 status register 1 reset value.
- #define _TIM2_SR2_RESET_VALUE ((uint8_t) 0x00)

  TIM2 status register 2 reset value.
- #define _TIM2_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Event generation register reset value.
- #define _TIM2_CCMR1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare mode register 1 reset value.
- #define _TIM2_CCMR2_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare mode register 2 reset value.

- #define _TIM2_CCER1_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Capture/compare enable register 1 reset value.
- #define _TIM2_CNTRH_RESET_VALUE ((uint8_t) 0x00)

  TIM2 counter register high byte reset value.
- #define _TIM2_CNTRL_RESET_VALUE ((uint8_t) 0x00)

  TIM2 counter register low byte reset value.
- #define _TIM2_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 clock prescaler register reset value.
- #define _TIM2_ARRH_RESET_VALUE ((uint8_t) 0xFF)

  TIM2 auto-reload register high byte reset value.
- #define _TIM2_ARRL_RESET_VALUE ((uint8_t) 0xFF)

  TIM2 auto-reload register low byte reset value.
- #define _TIM2_RCR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Repetition counter reset value.
- #define _TIM2_CCR1H_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 1 high byte reset value.
- #define _TIM2_CCR1L_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 1 low byte reset value.
- #define _TIM2_CCR2H_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 2 high byte reset value.
- #define _TIM2_CCR2L_RESET_VALUE ((uint8_t) 0x00)

  TIM2 16-bit capture/compare value 2 low byte reset value.
- #define _TIM2_BKR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Break register reset value.
- #define _TIM2_OISR_RESET_VALUE ((uint8_t) 0x00)

  TIM2 Output idle state register reset value.
- #define _TIM2_CEN ((uint8_t) (0x01 << 0))

  TIM2 Counter enable [0] (in _TIM2_CR1)
- #define _TIM2_UDIS ((uint8_t) (0x01 << 1))

  TIM2 Update disable [0] (in _TIM2_CR1)
- #define _TIM2_URS ((uint8_t) (0x01 << 2))

  TIM2 Update request source [0] (in _TIM2_CR1)
- #define _TIM2_OPM ((uint8_t) (0x01 << 3))

  TIM2 One-pulse mode [0] (in _TIM2_CR1)
- #define _TIM2_DIR ((uint8_t) (0x01 << 4))

  TIM2 Direction [0] (in _TIM2_CR1)
- #define _TIM2_ARPE ((uint8_t) (0x01 << 7))

  TIM2 Auto-reload preload enable [0] (in _TIM2_CR1)
- #define _TIM2_MMS ((uint8_t) (0x07 << 4))

  TIM2 Master mode selection [2:0] (in _TIM2_CR2)
- #define _TIM2_MMS0 ((uint8_t) (0x01 << 4))

  TIM2 Master mode selection [0] (in _TIM2_CR2)
- #define _TIM2_MMS1 ((uint8_t) (0x01 << 5))

  TIM2 Master mode selection [1] (in _TIM2_CR2)
- #define _TIM2_MMS2 ((uint8_t) (0x01 << 6))

  TIM2 Master mode selection [2] (in _TIM2_CR2)
- #define _TIM2_SMS ((uint8_t) (0x07 << 0))

  TIM2 Clock/trigger/slave mode selection [2:0] (in _TIM2_SMCR)
- #define _TIM2_SMS0 ((uint8_t) (0x01 << 0))

  TIM2 Clock/trigger/slave mode selection [0] (in _TIM2_SMCR)
- #define _TIM2_SMS1 ((uint8_t) (0x01 << 1))

TIM2 Clock/trigger/slave mode selection [1] (in _TIM2_SMCR)

- #define _TIM2_SMS2 ((uint8_t) (0x01 << 2))

  TIM2 Clock/trigger/slave mode selection [2] (in _TIM2_SMCR)
- #define _TIM2_TS ((uint8_t) (0x07 << 4))

  TIM2 Trigger selection [2:0] (in _TIM2_SMCR)
- #define _TIM2_TS0 ((uint8_t) (0x01 << 4))

  TIM2 Trigger selection [0] (in _TIM2_SMCR)
- #define _TIM2_TS1 ((uint8_t) (0x01 << 5))

  TIM2 Trigger selection [1] (in _TIM2_SMCR)
- #define _TIM2_TS2 ((uint8_t) (0x01 << 6))

  TIM2 Trigger selection [2] (in _TIM2_SMCR)
- #define _TIM2_MSM ((uint8_t) (0x01 << 7))

  TIM2 Master/slave mode [0] (in _TIM2_SMCR)
- #define _TIM2_ETF ((uint8_t) (0x0F << 0))

  TIM2 External trigger filter [3:0] (in _TIM2_ETR)
- #define _TIM2_ETF0 ((uint8_t) (0x01 << 0))

  TIM2 External trigger filter [0] (in _TIM2_ETR)
- #define _TIM2_ETF1 ((uint8_t) (0x01 << 1))

  TIM2 External trigger filter [1] (in _TIM2_ETR)
- #define _TIM2_ETF2 ((uint8_t) (0x01 << 2))

  TIM2 External trigger filter [2] (in _TIM2_ETR)
- #define _TIM2_ETF3 ((uint8_t) (0x01 << 3))

  TIM2 External trigger filter [3] (in _TIM2_ETR)
- #define _TIM2_ETPS ((uint8_t) (0x03 << 4))

  TIM2 External trigger prescaler [1:0] (in _TIM2_ETR)
- #define _TIM2_ETPS0 ((uint8_t) (0x01 << 4))

  TIM2 External trigger prescaler [0] (in _TIM2_ETR)
- #define _TIM2_ETPS1 ((uint8_t) (0x01 << 5))

  TIM2 External trigger prescaler [1] (in _TIM2_ETR)
- #define _TIM2_ECE ((uint8_t) (0x01 << 6))

  TIM2 External clock enable [0] (in _TIM2_ETR)
- #define _TIM2_ETP ((uint8_t) (0x01 << 7))

  TIM2 External trigger polarity [0] (in _TIM2_ETR)
- #define _TIM2_UIE ((uint8_t) (0x01 << 0))

  TIM2 Update interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_CC1IE ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_CC2IE ((uint8_t) (0x01 << 2))

  TIM2 Capture/compare 2 interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_TIE ((uint8_t) (0x01 << 6))

  TIM2 Trigger interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_BIE ((uint8_t) (0x01 << 7))

  TIM2 Break interrupt enable [0] (in _TIM2_IER)
- #define _TIM2_UIF ((uint8_t) (0x01 << 0))

  TIM2 Update interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC1IF ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC2IF ((uint8_t) (0x01 << 2))

  TIM2 Capture/compare 2 interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_TIF ((uint8_t) (0x01 << 6))

  TIM2 Trigger interrupt flag [0] (in _TIM2_SR1)

- #define _TIM2_BIF ((uint8_t) (0x01 << 7))
  TIM2 Break interrupt flag [0] (in _TIM2_SR1)
- #define _TIM2_CC1OF ((uint8_t) (0x01 << 1))
  TIM2 Capture/compare 1 overcapture flag [0] (in _TIM2_SR2)
- #define _TIM2_CC2OF ((uint8_t) (0x01 << 2))
  TIM2 Capture/compare 2 overcapture flag [0] (in _TIM2_SR2)
- #define _TIM2_UG ((uint8_t) (0x01 << 0))
  TIM2 Update generation [0] (in _TIM2_EGR)
- #define _TIM2_CC1G ((uint8_t) (0x01 << 1))
  TIM2 Capture/compare 1 generation [0] (in _TIM2_EGR)
- #define _TIM2_CC2G ((uint8_t) (0x01 << 2))
  TIM2 Capture/compare 2 generation [0] (in _TIM2_EGR)
- #define _TIM2_TG ((uint8_t) (0x01 << 6))
  TIM2 Trigger generation [0] (in _TIM2_EGR)
- #define _TIM2_BG ((uint8_t) (0x01 << 7))
  TIM2 Break generation [0] (in _TIM2_EGR)
- #define _TIM2_CC1S ((uint8_t) (0x03 << 0))
  TIM2 Compare 1 selection [1:0] (in _TIM2_CCMR1)
- #define _TIM2_CC1S0 ((uint8_t) (0x01 << 0))
  TIM2 Compare 1 selection [0] (in _TIM2_CCMR1)
- #define _TIM2_CC1S1 ((uint8_t) (0x01 << 1))
  TIM2 Compare 1 selection [1] (in _TIM2_CCMR1)
- #define _TIM2_OC1FE ((uint8_t) (0x01 << 2))
  TIM2 Output compare 1 fast enable [0] (in _TIM2_CCMR1)
- #define _TIM2_OC1PE ((uint8_t) (0x01 << 3))
  TIM2 Output compare 1 preload enable [0] (in _TIM2_CCMR1)
- #define _TIM2_OC1M ((uint8_t) (0x07 << 4))
  TIM2 Output compare 1 mode [2:0] (in _TIM2_CCMR1)
- #define _TIM2_OC1M0 ((uint8_t) (0x01 << 4))
  TIM2 Output compare 1 mode [0] (in _TIM2_CCMR1)
- #define _TIM2_OC1M1 ((uint8_t) (0x01 << 5))
  TIM2 Output compare 1 mode [1] (in _TIM2_CCMR1)
- #define _TIM2_OC1M2 ((uint8_t) (0x01 << 6))
  TIM2 Output compare 1 mode [2] (in _TIM2_CCMR1)
- #define _TIM2_IC1PSC ((uint8_t) (0x03 << 2))
  TIM2 Input capture 1 prescaler [1:0] (in _TIM2_CCMR1)
- #define _TIM2_IC1PSC0 ((uint8_t) (0x01 << 2))
  TIM2 Input capture 1 prescaler [0] (in _TIM2_CCMR1)
- #define _TIM2_IC1PSC1 ((uint8_t) (0x01 << 3))
  TIM2 Input capture 1 prescaler [1] (in _TIM2_CCMR1)
- #define _TIM2_IC1F ((uint8_t) (0x0F << 4))
  TIM2 Output compare 1 mode [3:0] (in _TIM2_CCMR1)
- #define _TIM2_IC1F0 ((uint8_t) (0x01 << 4))
  TIM2 Input capture 1 filter [0] (in _TIM2_CCMR1)
- #define _TIM2_IC1F1 ((uint8_t) (0x01 << 5))
  TIM2 Input capture 1 filter [1] (in _TIM2_CCMR1)
- #define _TIM2_IC1F2 ((uint8_t) (0x01 << 6))
  TIM2 Input capture 1 filter [2] (in _TIM2_CCMR1)
- #define _TIM2_IC1F3 ((uint8_t) (0x01 << 7))
  TIM2 Input capture 1 filter [3] (in _TIM2_CCMR1)
- #define _TIM2_CC2S ((uint8_t) (0x03 << 0))

TIM2 Compare 2 selection [1:0] (in _TIM2_CCMR2)

- #define _TIM2_CC2S0 ((uint8_t) (0x01 << 0))

  TIM2 Compare 2 selection [0] (in _TIM2_CCMR2)
- #define _TIM2_CC2S1 ((uint8_t) (0x01 << 1))

  TIM2 Compare 2 selection [1] (in _TIM2_CCMR2)
- #define _TIM2_OC2FE ((uint8_t) (0x01 << 2))

  TIM2 Output compare 2 fast enable [0] (in _TIM2_CCMR2)
- #define _TIM2_OC2PE ((uint8_t) (0x01 << 3))

  TIM2 Output compare 2 preload enable [0] (in _TIM2_CCMR2)
- #define _TIM2_OC2M ((uint8_t) (0x07 << 4))

  TIM2 Output compare 2 mode [2:0] (in _TIM2_CCMR2)
- #define _TIM2_OC2M0 ((uint8_t) (0x01 << 4))

  TIM2 Output compare 2 mode [0] (in _TIM2_CCMR2)
- #define _TIM2_OC2M1 ((uint8_t) (0x01 << 5))

  TIM2 Output compare 2 mode [1] (in _TIM2_CCMR2)
- #define _TIM2_OC2M2 ((uint8_t) (0x01 << 6))

  TIM2 Output compare 2 mode [2] (in _TIM2_CCMR2)
- #define _TIM2_IC2PSC ((uint8_t) (0x03 << 2))

  TIM2 Input capture 2 prescaler [1:0] (in _TIM2_CCMR2)
- #define _TIM2_IC2PSC0 ((uint8_t) (0x01 << 2))

  TIM2 Input capture 2 prescaler [0] (in _TIM2_CCMR2)
- #define _TIM2_IC2PSC1 ((uint8_t) (0x01 << 3))

  TIM2 Input capture 2 prescaler [1] (in _TIM2_CCMR2)
- #define _TIM2_IC2F ((uint8_t) (0x0F << 4))

  TIM2 Output compare 2 mode [3:0] (in _TIM2_CCMR2)
- #define _TIM2_IC2F0 ((uint8_t) (0x01 << 4))

  TIM2 Input capture 2 filter [0] (in _TIM2_CCMR2)
- #define _TIM2_IC2F1 ((uint8_t) (0x01 << 5))

  TIM2 Input capture 2 filter [1] (in _TIM2_CCMR2)
- #define _TIM2_IC2F2 ((uint8_t) (0x01 << 6))

  TIM2 Input capture 2 filter [2] (in _TIM2_CCMR2)
- #define _TIM2_IC2F3 ((uint8_t) (0x01 << 7))

  TIM2 Input capture 2 filter [3] (in _TIM2_CCMR2)
- #define _TIM2_CC1E ((uint8_t) (0x01 << 0))

  TIM2 Capture/compare 1 output enable [0] (in _TIM2_CCER1)
- #define _TIM2_CC1P ((uint8_t) (0x01 << 1))

  TIM2 Capture/compare 1 output polarity [0] (in _TIM2_CCER1)
- #define _TIM2_CC2E ((uint8_t) (0x01 << 4))

  TIM2 Capture/compare 2 output enable [0] (in _TIM2_CCER1)
- #define _TIM2_CC2P ((uint8_t) (0x01 << 5))

  TIM2 Capture/compare 2 output polarity [0] (in _TIM2_CCER1)
- #define _TIM2_PSC ((uint8_t) (0x07 << 0))

  TIM2 prescaler [2:0] (in _TIM2_PSCR)
- #define _TIM2_PSC0 ((uint8_t) (0x01 << 0))

  TIM2 prescaler [0] (in _TIM2_PSCR)
- #define _TIM2_PSC1 ((uint8_t) (0x01 << 1))

  TIM2 prescaler [1] (in _TIM2_PSCR)
- #define _TIM2_PSC2 ((uint8_t) (0x01 << 2))

  TIM2 prescaler [2] (in _TIM2_PSCR)
- #define _TIM2_LOCK ((int8_t) (0x03 << 0))

  TIM2 Lock configuration [1:0] (in _TIM2_BKR)

- #define _TIM2_LOCK0 ((uint8_t) (0x01 << 0))
  TIM2 Lock configuration [0] (in _TIM2_BKR)
- #define _TIM2_LOCK1 ((uint8_t) (0x01 << 1))
  TIM2 Lock configuration [1] (in _TIM2_BKR)
- #define _TIM2_OSSI ((uint8_t) (0x01 << 2))
  TIM2 Off state selection for idle mode [0] (in _TIM2_BKR)
- #define _TIM2_BKE ((uint8_t) (0x01 << 4))
  TIM2 Break enable [0] (in _TIM2_BKR)
- #define _TIM2_BKP ((uint8_t) (0x01 << 5))
  TIM2 Break polarity [0] (in _TIM2_BKR)
- #define _TIM2_AOE ((uint8_t) (0x01 << 6))
  TIM2 Automatic output enable [0] (in _TIM2_BKR)
- #define _TIM2_MOE ((uint8_t) (0x01 << 7))
  TIM2 Main output enable [0] (in _TIM2_BKR)
- #define _TIM2_OIS1 ((uint8_t) (0x01 << 0))
  TIM2 Output idle state 1 (OC1 output) [0] (in _TIM2_OISR)
- #define _TIM2_OIS2 ((uint8_t) (0x01 << 2))
  TIM2 Output idle state 2 (OC2 output) [0] (in _TIM2_OISR)
- #define _TIM3 _SFR(TIM3_3_t, TIM3_AddressBase)
  TIM3 struct/bit access.
- #define _TIM3_CR1 _SFR(uint8_t, TIM3_AddressBase+0x00)
  TIM3 control register 1.
- #define _TIM3_CR2 _SFR(uint8_t, TIM3_AddressBase+0x01)
  TIM3 control register 2.
- #define _TIM3_SMCR _SFR(uint8_t, TIM3_AddressBase+0x02)
  TIM3 Slave mode control register.
- #define _TIM3_ETR _SFR(uint8_t, TIM3_AddressBase+0x03)
  TIM3 External trigger register.
- #define _TIM3_IER _SFR(uint8_t, TIM3_AddressBase+0x04)
  TIM3 interrupt enable register.
- #define _TIM3_SR1 _SFR(uint8_t, TIM3_AddressBase+0x05)
  TIM3 status register 1.
- #define _TIM3_SR2 _SFR(uint8_t, TIM3_AddressBase+0x06)
  TIM3 status register 2.
- #define _TIM3_EGR _SFR(uint8_t, TIM3_AddressBase+0x07)
  TIM3 Event generation register.
- #define _TIM3_CCMR1 _SFR(uint8_t, TIM3_AddressBase+0x08)
  TIM3 Capture/compare mode register 1.
- #define _TIM3_CCMR2 _SFR(uint8_t, TIM3_AddressBase+0x09)
  TIM3 Capture/compare mode register 2.
- #define _TIM3_CCER1 _SFR(uint8_t, TIM3_AddressBase+0x0A)
  TIM3 Capture/compare enable register 1.
- #define _TIM3_CNTRH _SFR(uint8_t, TIM3_AddressBase+0x0B)
  TIM3 counter register high byte.
- #define _TIM3_CNTRL _SFR(uint8_t, TIM3_AddressBase+0x0C)
  TIM3 counter register low byte.
- #define _TIM3_PSCR _SFR(uint8_t, TIM3_AddressBase+0x0D)
  TIM3 clock prescaler register.
- #define _TIM3_ARRH _SFR(uint8_t, TIM3_AddressBase+0x0E)
  TIM3 auto-reload register high byte.
- #define _TIM3_ARRL _SFR(uint8_t, TIM3_AddressBase+0x0F)

TIM3 auto-reload register low byte.

- #define _TIM3_CCR1H _SFR(uint8_t, TIM3_AddressBase+0x10)

  TIM3 16-bit capture/compare value 1 high byte.
- #define _TIM3_CCR1L _SFR(uint8_t, TIM3_AddressBase+0x11)

  TIM3 16-bit capture/compare value 1 low byte.
- #define _TIM3_CCR2H _SFR(uint8_t, TIM3_AddressBase+0x12)

  TIM3 16-bit capture/compare value 2 high byte.
- #define _TIM3_CCR2L _SFR(uint8_t, TIM3_AddressBase+0x13)

  TIM3 16-bit capture/compare value 2 low byte.
- #define _TIM3_BKR _SFR(uint8_t, TIM3_AddressBase+0x14)

  TIM3 Break register.
- #define _TIM3_OISR _SFR(uint8_t, TIM3_AddressBase+0x15)

  TIM3 Output idle state register.
- #define _TIM3_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 control register 1 reset value.
- #define _TIM3_CR2_RESET_VALUE ((uint8_t) 0x00)

  TIM3 control register 2 reset value.
- #define _TIM3_SMCR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Slave mode control register reset value.
- #define _TIM3_ETR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 External trigger register reset value.
- #define _TIM3_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM3 interrupt enable register reset value.
- #define _TIM3_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 status register 1 reset value.
- #define _TIM3_SR2_RESET_VALUE ((uint8_t) 0x00)

  TIM3 status register 2 reset value.
- #define _TIM3_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Event generation register reset value.
- #define _TIM3_CCMR1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Capture/compare mode register 1 reset value.
- #define _TIM3_CCMR2_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Capture/compare mode register 2 reset value.
- #define _TIM3_CCER1_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Capture/compare enable register 1 reset value.
- #define _TIM3_CNTRH_RESET_VALUE ((uint8_t) 0x00)

  TIM3 counter register high byte reset value.
- #define _TIM3_CNTRL_RESET_VALUE ((uint8_t) 0x00)

  TIM3 counter register low byte reset value.
- #define _TIM3_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 clock prescaler register reset value.
- #define _TIM3_ARRH_RESET_VALUE ((uint8_t) 0xFF)

  TIM3 auto-reload register high byte reset value.
- #define _TIM3_ARRL_RESET_VALUE ((uint8_t) 0xFF)

  TIM3 auto-reload register low byte reset value.
- #define _TIM3_RCR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Repetition counter reset value.
- #define _TIM3_CCR1H_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 1 high byte reset value.
- #define _TIM3_CCR1L_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 1 low byte reset value.

- #define _TIM3_CCR2H_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 2 high byte reset value.
- #define _TIM3_CCR2L_RESET_VALUE ((uint8_t) 0x00)

  TIM3 16-bit capture/compare value 2 low byte reset value.
- #define _TIM3_BKR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Break register reset value.
- #define _TIM3_OISR_RESET_VALUE ((uint8_t) 0x00)

  TIM3 Output idle state register reset value.
- #define _TIM3_CEN ((uint8_t) (0x01 << 0))

  TIM3 Counter enable [0] (in _TIM3_CR1)
- #define _TIM3_UDIS ((uint8_t) (0x01 << 1))

  TIM3 Update disable [0] (in _TIM3_CR1)
- #define _TIM3_URS ((uint8_t) (0x01 << 2))

  TIM3 Update request source [0] (in _TIM3_CR1)
- #define _TIM3_OPM ((uint8_t) (0x01 << 3))

  TIM3 One-pulse mode [0] (in _TIM3_CR1)
- #define _TIM3_DIR ((uint8_t) (0x01 << 4))

  TIM3 Direction [0] (in _TIM3_CR1)
- #define _TIM3_ARPE ((uint8_t) (0x01 << 7))

  TIM3 Auto-reload preload enable [0] (in _TIM3_CR1)
- #define _TIM3_MMS ((uint8_t) (0x07 << 4))

  TIM3 Master mode selection [2:0] (in _TIM3_CR2)
- #define _TIM3_MMS0 ((uint8_t) (0x01 << 4))

  TIM3 Master mode selection [0] (in _TIM3_CR2)
- #define _TIM3_MMS1 ((uint8_t) (0x01 << 5))

  TIM3 Master mode selection [1] (in _TIM3_CR2)
- #define _TIM3_MMS2 ((uint8_t) (0x01 << 6))

  TIM3 Master mode selection [2] (in _TIM3_CR2)
- #define _TIM3_SMS ((uint8_t) (0x07 << 0))

  TIM3 Clock/trigger/slave mode selection [2:0] (in _TIM3_SMCR)
- #define _TIM3_SMS0 ((uint8_t) (0x01 << 0))

  TIM3 Clock/trigger/slave mode selection [0] (in _TIM3_SMCR)
- #define _TIM3_SMS1 ((uint8_t) (0x01 << 1))

  TIM3 Clock/trigger/slave mode selection [1] (in _TIM3_SMCR)
- #define _TIM3_SMS2 ((uint8_t) (0x01 << 2))

  TIM3 Clock/trigger/slave mode selection [2] (in _TIM3_SMCR)
- #define _TIM3_TS ((uint8_t) (0x07 << 4))

  TIM3 Trigger selection [2:0] (in _TIM3_SMCR)
- #define _TIM3_TS0 ((uint8_t) (0x01 << 4))

  TIM3 Trigger selection [0] (in _TIM3_SMCR)
- #define _TIM3_TS1 ((uint8_t) (0x01 << 5))

  TIM3 Trigger selection [1] (in _TIM3_SMCR)
- #define _TIM3_TS2 ((uint8_t) (0x01 << 6))

  TIM3 Trigger selection [2] (in _TIM3_SMCR)
- #define _TIM3_MSM ((uint8_t) (0x01 << 7))

  TIM3 Master/slave mode [0] (in _TIM3_SMCR)
- #define _TIM3_ETF ((uint8_t) (0x0F << 0))

  TIM3 External trigger filter [3:0] (in _TIM3_ETR)
- #define _TIM3_ETF0 ((uint8_t) (0x01 << 0))

  TIM3 External trigger filter [0] (in _TIM3_ETR)
- #define _TIM3_ETF1 ((uint8_t) (0x01 << 1))

TIM3 External trigger filter [1] (in _TIM3_ETR)

- #define _TIM3_ETF2 ((uint8_t) (0x01 << 2))

  TIM3 External trigger filter [2] (in _TIM3_ETR)
- #define _TIM3_ETF3 ((uint8_t) (0x01 << 3))

  TIM3 External trigger filter [3] (in _TIM3_ETR)
- #define _TIM3_ETPS ((uint8_t) (0x03 << 4))

  TIM3 External trigger prescaler [1:0] (in _TIM3_ETR)
- #define _TIM3_ETPS0 ((uint8_t) (0x01 << 4))

  TIM3 External trigger prescaler [0] (in _TIM3_ETR)
- #define _TIM3_ETPS1 ((uint8_t) (0x01 << 5))

  TIM3 External trigger prescaler [1] (in _TIM3_ETR)
- #define _TIM3_ECE ((uint8_t) (0x01 << 6))

  TIM3 External clock enable [0] (in _TIM3_ETR)
- #define _TIM3_ETP ((uint8_t) (0x01 << 7))

  TIM3 External trigger polarity [0] (in _TIM3_ETR)
- #define _TIM3_UIE ((uint8_t) (0x01 << 0))

  TIM3 Update interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_CC1IE ((uint8_t) (0x01 << 1))

  TIM3 Capture/compare 1 interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_CC2IE ((uint8_t) (0x01 << 2))

  TIM3 Capture/compare 2 interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_TIE ((uint8_t) (0x01 << 6))

  TIM3 Trigger interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_BIE ((uint8_t) (0x01 << 7))

  TIM3 Break interrupt enable [0] (in _TIM3_IER)
- #define _TIM3_UIF ((uint8_t) (0x01 << 0))

  TIM3 Update interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_CC1IF ((uint8_t) (0x01 << 1))

  TIM3 Capture/compare 1 interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_CC2IF ((uint8_t) (0x01 << 2))

  TIM3 Capture/compare 2 interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_TIF ((uint8_t) (0x01 << 6))

  TIM3 Trigger interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_BIF ((uint8_t) (0x01 << 7))

  TIM3 Break interrupt flag [0] (in _TIM3_SR1)
- #define _TIM3_CC1OF ((uint8_t) (0x01 << 1))

  TIM3 Capture/compare 1 overcapture flag [0] (in _TIM3_SR2)
- #define _TIM3_CC2OF ((uint8_t) (0x01 << 2))

  TIM3 Capture/compare 2 overcapture flag [0] (in _TIM3_SR2)
- #define _TIM3_UG ((uint8_t) (0x01 << 0))

  TIM3 Update generation [0] (in _TIM3_EGR)
- #define _TIM3_CC1G ((uint8_t) (0x01 << 1))

  TIM3 Capture/compare 1 generation [0] (in _TIM3_EGR)
- #define _TIM3_CC2G ((uint8_t) (0x01 << 2))

  TIM3 Capture/compare 2 generation [0] (in _TIM3_EGR)
- #define _TIM3_TG ((uint8_t) (0x01 << 6))

  TIM3 Trigger generation [0] (in _TIM3_EGR)
- #define _TIM3_BG ((uint8_t) (0x01 << 7))

  TIM3 Break generation [0] (in _TIM3_EGR)
- #define _TIM3_CC1S ((uint8_t) (0x03 << 0))

  TIM3 Compare 1 selection [1:0] (in _TIM3_CCMR1)

- #define _TIM3_CC1S0 ((uint8_t) (0x01 << 0))
  TIM3 Compare 1 selection [0] (in _TIM3_CCMR1)
- #define _TIM3_CC1S1 ((uint8_t) (0x01 << 1))
  TIM3 Compare 1 selection [1] (in _TIM3_CCMR1)
- #define _TIM3_OC1FE ((uint8_t) (0x01 << 2))
  TIM3 Output compare 1 fast enable [0] (in _TIM3_CCMR1)
- #define _TIM3_OC1PE ((uint8_t) (0x01 << 3))
  TIM3 Output compare 1 preload enable [0] (in _TIM3_CCMR1)
- #define _TIM3_OC1M ((uint8_t) (0x07 << 4))
  TIM3 Output compare 1 mode [2:0] (in _TIM3_CCMR1)
- #define _TIM3_OC1M0 ((uint8_t) (0x01 << 4))
  TIM3 Output compare 1 mode [0] (in _TIM3_CCMR1)
- #define _TIM3_OC1M1 ((uint8_t) (0x01 << 5))
  TIM3 Output compare 1 mode [1] (in _TIM3_CCMR1)
- #define _TIM3_OC1M2 ((uint8_t) (0x01 << 6))
  TIM3 Output compare 1 mode [2] (in _TIM3_CCMR1)
- #define _TIM3_IC1PSC ((uint8_t) (0x03 << 2))
  TIM3 Input capture 1 prescaler [1:0] (in _TIM3_CCMR1)
- #define _TIM3_IC1PSC0 ((uint8_t) (0x01 << 2))
  TIM3 Input capture 1 prescaler [0] (in _TIM3_CCMR1)
- #define _TIM3_IC1PSC1 ((uint8_t) (0x01 << 3))
  TIM3 Input capture 1 prescaler [1] (in _TIM3_CCMR1)
- #define _TIM3_IC1F ((uint8_t) (0x0F << 4))
  TIM3 Output compare 1 mode [3:0] (in _TIM3_CCMR1)
- #define _TIM3_IC1F0 ((uint8_t) (0x01 << 4))
  TIM3 Input capture 1 filter [0] (in _TIM3_CCMR1)
- #define _TIM3_IC1F1 ((uint8_t) (0x01 << 5))
  TIM3 Input capture 1 filter [1] (in _TIM3_CCMR1)
- #define _TIM3_IC1F2 ((uint8_t) (0x01 << 6))
  TIM3 Input capture 1 filter [2] (in _TIM3_CCMR1)
- #define _TIM3_IC1F3 ((uint8_t) (0x01 << 7))
  TIM3 Input capture 1 filter [3] (in _TIM3_CCMR1)
- #define _TIM3_CC2S ((uint8_t) (0x03 << 0))
  TIM3 Compare 2 selection [1:0] (in _TIM3_CCMR2)
- #define _TIM3_CC2S0 ((uint8_t) (0x01 << 0))
  TIM3 Compare 2 selection [0] (in _TIM3_CCMR2)
- #define _TIM3_CC2S1 ((uint8_t) (0x01 << 1))
  TIM3 Compare 2 selection [1] (in _TIM3_CCMR2)
- #define _TIM3_OC2FE ((uint8_t) (0x01 << 2))
  TIM3 Output compare 2 fast enable [0] (in _TIM3_CCMR2)
- #define _TIM3_OC2PE ((uint8_t) (0x01 << 3))
  TIM3 Output compare 2 preload enable [0] (in _TIM3_CCMR2)
- #define _TIM3_OC2M ((uint8_t) (0x07 << 4))
  TIM3 Output compare 2 mode [2:0] (in _TIM3_CCMR2)
- #define _TIM3_OC2M0 ((uint8_t) (0x01 << 4))
  TIM3 Output compare 2 mode [0] (in _TIM3_CCMR2)
- #define _TIM3_OC2M1 ((uint8_t) (0x01 << 5))
  TIM3 Output compare 2 mode [1] (in _TIM3_CCMR2)
- #define _TIM3_OC2M2 ((uint8_t) (0x01 << 6))
  TIM3 Output compare 2 mode [2] (in _TIM3_CCMR2)
- #define _TIM3_IC2PSC ((uint8_t) (0x03 << 2))

TIM3 Input capture 2 prescaler [1:0] (in _TIM3_CCMR2)

- #define _TIM3_IC2PSC0 ((uint8_t) (0x01 << 2))

  TIM3 Input capture 2 prescaler [0] (in _TIM3_CCMR2)
- #define _TIM3_IC2PSC1 ((uint8_t) (0x01 << 3))

  TIM3 Input capture 2 prescaler [1] (in _TIM3_CCMR2)
- #define _TIM3_IC2F ((uint8_t) (0x0F << 4))

  TIM3 Output compare 2 mode [3:0] (in _TIM3_CCMR2)
- #define _TIM3_IC2F0 ((uint8_t) (0x01 << 4))

  TIM3 Input capture 2 filter [0] (in _TIM3_CCMR2)
- #define _TIM3_IC2F1 ((uint8_t) (0x01 << 5))

  TIM3 Input capture 2 filter [1] (in _TIM3_CCMR2)
- #define _TIM3_IC2F2 ((uint8_t) (0x01 << 6))

  TIM3 Input capture 2 filter [2] (in _TIM3_CCMR2)
- #define _TIM3_IC2F3 ((uint8_t) (0x01 << 7))

  TIM3 Input capture 2 filter [3] (in _TIM3_CCMR2)
- #define _TIM3_CC1E ((uint8_t) (0x01 << 0))

  TIM3 Capture/compare 1 output enable [0] (in _TIM3_CCER1)
- #define _TIM3_CC1P ((uint8_t) (0x01 << 1))

  TIM3 Capture/compare 1 output polarity [0] (in _TIM3_CCER1)
- #define _TIM3_CC2E ((uint8_t) (0x01 << 4))

  TIM3 Capture/compare 2 output enable [0] (in _TIM3_CCER1)
- #define _TIM3_CC2P ((uint8_t) (0x01 << 5))

  TIM3 Capture/compare 2 output polarity [0] (in _TIM3_CCER1)
- #define _TIM3_PSC ((uint8_t) (0x07 << 0))

  TIM3 prescaler [2:0] (in _TIM3_PSCR)
- #define _TIM3_PSC0 ((uint8_t) (0x01 << 0))

  TIM3 prescaler [0] (in _TIM3_PSCR)
- #define _TIM3_PSC1 ((uint8_t) (0x01 << 1))

  TIM3 prescaler [1] (in _TIM3_PSCR)
- #define _TIM3_PSC2 ((uint8_t) (0x01 << 2))

  TIM3 prescaler [2] (in _TIM3_PSCR)
- #define _TIM3_LOCK ((int8_t) (0x03 << 0))

  TIM3 Lock configuration [1:0] (in _TIM3_BKR)
- #define _TIM3_LOCK0 ((uint8_t) (0x01 << 0))

  TIM3 Lock configuration [0] (in _TIM3_BKR)
- #define _TIM3_LOCK1 ((uint8_t) (0x01 << 1))

  TIM3 Lock configuration [1] (in _TIM3_BKR)
- #define _TIM3_OSSI ((uint8_t) (0x01 << 2))

  TIM3 Off state selection for idle mode [0] (in _TIM3_BKR)
- #define _TIM3_BKE ((uint8_t) (0x01 << 4))

  TIM3 Break enable [0] (in _TIM3_BKR)
- #define _TIM3_BKP ((uint8_t) (0x01 << 5))

  TIM3 Break polarity [0] (in _TIM3_BKR)
- #define _TIM3_AOE ((uint8_t) (0x01 << 6))

  TIM3 Automatic output enable [0] (in _TIM3_BKR)
- #define _TIM3_MOE ((uint8_t) (0x01 << 7))

  TIM3 Main output enable [0] (in _TIM3_BKR)
- #define _TIM3_OIS1 ((uint8_t) (0x01 << 0))

  TIM3 Output idle state 1 (OC1 output) [0] (in _TIM3_OISR)
- #define _TIM3_OIS2 ((uint8_t) (0x01 << 2))

  TIM3 Output idle state 2 (OC2 output) [0] (in _TIM3_OISR)

- #define _TIM4 _SFR(TIM4_t, TIM4_AddressBase)

  TIM4 struct/bit access.
- #define _TIM4_CR1 _SFR(uint8_t, TIM4_AddressBase+0x00)

  TIM4 control register 1.
- #define _TIM4_CR2 _SFR(uint8_t, TIM4_AddressBase+0x01)

  TIM4 control register 2.
- #define _TIM4_SMCR _SFR(uint8_t, TIM4_AddressBase+0x02)

  TIM4 Slave mode control register.
- #define _TIM4_IER _SFR(uint8_t, TIM4_AddressBase+0x03)

  TIM4 interrupt enable register.
- #define _TIM4_SR1 _SFR(uint8_t, TIM4_AddressBase+0x04)

  TIM4 status register.
- #define _TIM4_EGR _SFR(uint8_t, TIM4_AddressBase+0x05)

  TIM4 event generation register.
- #define _TIM4_CNTR _SFR(uint8_t, TIM4_AddressBase+0x06)

  TIM4 counter register.
- #define _TIM4_PSCR _SFR(uint8_t, TIM4_AddressBase+0x07)

  TIM4 clock prescaler register.
- #define _TIM4_ARR _SFR(uint8_t, TIM4_AddressBase+0x08)

  TIM4 auto-reload register.
- #define _TIM4_CR1_RESET_VALUE ((uint8_t) 0x00)

  TIM4 control register 1 reset value.
- #define _TIM4_CR2_RESET_VALUE ((uint8_t) 0x00)

  TIM4 control register 2 reset value.
- #define _TIM4_SMCR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 Slave mode control register reset value.
- #define _TIM4_IER_RESET_VALUE ((uint8_t) 0x00)

  TIM4 interrupt enable register reset value.
- #define _TIM4_SR1_RESET_VALUE ((uint8_t) 0x00)

  TIM4 status register reset value.
- #define _TIM4_EGR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 event generation register reset value.
- #define _TIM4_CNTR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 counter register reset value.
- #define _TIM4_PSCR_RESET_VALUE ((uint8_t) 0x00)

  TIM4 clock prescaler register reset value.
- #define _TIM4_ARR_RESET_VALUE ((uint8_t) 0xFF)

  TIM4 auto-reload register reset value.
- #define _TIM4_CEN ((uint8_t) (0x01 << 0))

  TIM4 Counter enable [0] (in _TIM4_CR1)
- #define _TIM4_UDIS ((uint8_t) (0x01 << 1))

  TIM4 Update disable [0] (in _TIM4_CR1)
- #define _TIM4_URS ((uint8_t) (0x01 << 2))

  TIM4 Update request source [0] (in _TIM4_CR1)
- #define _TIM4_OPM ((uint8_t) (0x01 << 3))

  TIM4 One-pulse mode [0] (in _TIM4_CR1)
- #define _TIM4_ARPE ((uint8_t) (0x01 << 7))

  TIM4 Auto-reload preload enable [0] (in _TIM4_CR)
- #define _TIM4_MMS ((uint8_t) (0x07 << 4))

  TIM4 Master mode selection [2:0] (in _TIM4_CR2)
- #define _TIM4_MMS0 ((uint8_t) (0x01 << 4))

TIM4 Master mode selection [0] (in _TIM4_CR2)

- #define _TIM4_MMS1 ((uint8_t) (0x01 << 5))

  TIM4 Master mode selection [1] (in _TIM4_CR2)
- #define _TIM4_MMS2 ((uint8_t) (0x01 << 6))

  TIM4 Master mode selection [2] (in _TIM4_CR2)
- #define _TIM4_SMS ((uint8_t) (0x07 << 0))

  TIM4 Clock/trigger/slave mode selection [2:0] (in _TIM4_SMCR)
- #define _TIM4_SMS0 ((uint8_t) (0x01 << 0))

  TIM4 Clock/trigger/slave mode selection [0] (in _TIM4_SMCR)
- #define _TIM4_SMS1 ((uint8_t) (0x01 << 1))

  TIM4 Clock/trigger/slave mode selection [1] (in _TIM4_SMCR)
- #define _TIM4_SMS2 ((uint8_t) (0x01 << 2))

  TIM4 Clock/trigger/slave mode selection [2] (in _TIM4_SMCR)
- #define _TIM4_TS ((uint8_t) (0x07 << 4))

  TIM4 Trigger selection [2:0] (in _TIM4_SMCR)
- #define _TIM4_TS0 ((uint8_t) (0x01 << 4))

  TIM4 Trigger selection [0] (in _TIM4_SMCR)
- #define _TIM4_TS1 ((uint8_t) (0x01 << 5))

  TIM4 Trigger selection [1] (in _TIM4_SMCR)
- #define _TIM4_TS2 ((uint8_t) (0x01 << 6))

  TIM4 Trigger selection [2] (in _TIM4_SMCR)
- #define _TIM4_MSM ((uint8_t) (0x01 << 7))

  TIM4 Master/slave mode [0] (in _TIM4_SMCR)
- #define _TIM4_UIE ((uint8_t) (0x01 << 0))

  TIM4 Update interrupt enable [0] (in _TIM4_IER)
- #define _TIM4_TIE ((uint8_t) (0x01 << 6))

  TIM4 Trigger interrupt enable [0] (in _TIM4_IER)
- #define _TIM4_UIF ((uint8_t) (0x01 << 0))

  TIM4 Update interrupt flag [0] (in _TIM4_SR1)
- #define _TIM4_TIF ((uint8_t) (0x01 << 6))

  TIM4 Trigger interrupt flag [0] (in _TIM4_SR1)
- #define _TIM4_UG ((uint8_t) (0x01 << 0))

  TIM4 Update generation [0] (in _TIM4_EGR)
- #define _TIM4_TG ((uint8_t) (0x01 << 6))

  TIM4 Trigger generation [0] (in _TIM4_EGR)
- #define _TIM4_PSC ((uint8_t) (0x0F << 0))

  TIM4 clock prescaler [3:0] (in _TIM4_PSCR)
- #define _TIM4_PSC0 ((uint8_t) (0x01 << 0))

  TIM4 clock prescaler [0] (in _TIM4_PSCR)
- #define _TIM4_PSC1 ((uint8_t) (0x01 << 1))

  TIM4 clock prescaler [1] (in _TIM4_PSCR)
- #define _TIM4_PSC2 ((uint8_t) (0x01 << 2))

  TIM4 clock prescaler [2] (in _TIM4_PSCR)
- #define _TIM4_PSC3 ((uint8_t) (0x01 << 3))

  TIM4 clock prescaler [3] (in _TIM4_PSCR)
- #define _IRTIM _SFR(IRTIM_t, IRTIM_AddressBase)

  IRTIM struct/bit access.
- #define _IRTIM_CR1 _SFR(uint8_t, IRTIM_AddressBase+0x00)

  IRTIM control register.
- #define _IRTIM_CR_RESET_VALUE ((uint8_t) 0x00)

  IRTIM control register reset value.

- #define _IRTIM_IR_EN ((uint8_t) (0x01 << 0))

  IRTIM Infrared output enable [0] (in _IRTIM_CR)
- #define _IRTIM_HS_EN ((uint8_t) (0x01 << 1))

  IRTIM High Sink LED driver enable [0] (in _IRTIM_CR)
- #define _COMP _SFR(COMP_t_t, COMP_AddressBase)

  COMP struct/bit access.
- #define _COMP_CR _SFR(uint8_t, COMP_AddressBase+0x00)

  Comparator control register.
- #define _COMP_CSR _SFR(uint8_t, COMP_AddressBase+0x01)

  Comparator control status register.
- #define _COMP_CCS _SFR(uint8_t, COMP_AddressBase+0x02)

  Comparator channel selection.
- #define _COMP_CR_RESET_VALUE ((uint8_t) 0x00)

  Comparator control register reset value.
- #define _COMP_CSR_RESET_VALUE ((uint8_t) 0x00)

  Comparator control status register reset value.
- #define _COMP_CCS_RESET_VALUE ((uint8_t) 0x00)

  Comparator channel selection reset value.
- #define _COMP_BIAS_EN ((uint8_t) (0x01 << 0))

  COMP Bias enable [0] (in _COMP_CR)
- #define _COMP_COMP1_EN ((uint8_t) (0x01 << 1))

  COMP First comparator enable [0] (in _COMP_CR)
- #define _COMP_COMP2_EN ((uint8_t) (0x01 << 2))

  COMP Second comparator enable [0] (in _COMP_CR)
- #define _COMP_COMPREF ((uint8_t) (0x01 << 3))

  COMP Comparator reference [0] (in _COMP_CR)
- #define _COMP_POL ((uint8_t) (0x01 << 4))

  COMP Comparator polarity [0] (in _COMP_CR)
- #define _COMP_CNF_TIM ((uint8_t) (0x03 << 5))

  COMP Comparator 1/2 output connected to TIM2/3 capture or break [1:0] (in _COMP_CR)
- #define _COMP_CNF_TIM0 ((uint8_t) (0x03 << 5))

  COMP Comparator 1/2 output connected to TIM2/3 capture or break [0] (in _COMP_CR)
- #define _COMP_CNF_TIM1 ((uint8_t) (0x03 << 6))

  COMP Comparator 1/2 output connected to TIM2/3 capture or break [1] (in _COMP_CR)
- #define _COMP_IC1_BK ((uint8_t) (0x01 << 7))

  COMP Input capture 1 / break selection [0] (in _COMP_CR)
- #define _COMP_COMP1_OUT ((uint8_t) (0x01 << 0))

  COMP First comparator output [0] (in _COMP_CSR)
- #define _COMP_COMP2_OUT ((uint8_t) (0x01 << 1))

  COMP Second comparator output [0] (in _COMP_CSR)
- #define _COMP_CEF1 ((uint8_t) (0x01 << 4))

  COMP First comparator event flag [0] (in _COMP_CSR)
- #define _COMP_ITEN1 ((uint8_t) (0x01 << 5))

  COMP First comparator interrupt enable [0] (in _COMP_CSR)
- #define _COMP_CEF2 ((uint8_t) (0x01 << 6))

  COMP Second comparator event flag [0] (in _COMP_CSR)
- #define _COMP_ITEN2 ((uint8_t) (0x01 << 7))

  COMP Second comparator interrupt enable [0] (in _COMP_CSR)
- #define _COMP_COMP1_CH1 ((uint8_t) (0x01 << 0))

  COMP Comparator 1 switch 1 enable [0] (in _COMP_CCS)
- #define _COMP_COMP1_CH2 ((uint8_t) (0x01 << 1))

COMP Comparator 1 switch 2 enable [0] (in _COMP_CCS)

- #define _COMP_COMP1_CH3 ((uint8_t) (0x01 << 2))

  COMP Comparator 1 switch 3 enable [0] (in _COMP_CCS)
- #define _COMP_COMP1_CH4 ((uint8_t) (0x01 << 3))

  COMP Comparator 1 switch 4 enable [0] (in _COMP_CCS)
- #define _COMP_COMP2_CH1 ((uint8_t) (0x01 << 4))

  COMP Comparator 2 switch 1 enable [0] (in _COMP_CCS)
- #define _COMP_COMP2_CH2 ((uint8_t) (0x01 << 5))

  COMP Comparator 2 switch 2 enable [0] (in _COMP_CCS)
- #define _COMP_COMP2_CH3 ((uint8_t) (0x01 << 6))

  COMP Comparator 2 switch 3 enable [0] (in _COMP_CCS)
- #define _COMP_COMP2_CH4 ((uint8_t) (0x01 << 7))

  COMP Comparator 2 switch 4 enable [0] (in _COMP_CCS)
- #define _CFG _SFR(CFG_t, CFG_AddressBase)

  CFG struct/bit access.
- #define _CFG_GCR _SFR(uint8_t, CFG_AddressBase+0x00)

  Global configuration register (CFG_GCR)
- #define _CFG_GCR_RESET_VALUE ((uint8_t)0x00)
- #define _CFG_SWD ((uint8_t) (0x01 << 0))

  SWIM disable [0].
- #define _CFG_AL ((uint8_t) (0x01 << 1))

  Activation level [0].
- #define _ITC _SFR(ITC_t, ITC_AddressBase)

  ITC struct/bit access.
- #define _ITC_SPR1 _SFR(uint8_t, ITC_AddressBase+0x00)

  Interrupt priority register 1/8.
- #define _ITC_SPR2 _SFR(uint8_t, ITC_AddressBase+0x01)

  Interrupt priority register 2/8.
- #define _ITC_SPR3 _SFR(uint8_t, ITC_AddressBase+0x02)

  Interrupt priority register 3/8.
- #define _ITC_SPR4 _SFR(uint8_t, ITC_AddressBase+0x03)

  Interrupt priority register 4/8.
- #define _ITC_SPR5 _SFR(uint8_t, ITC_AddressBase+0x04)

  Interrupt priority register 5/8.
- #define _ITC_SPR6 _SFR(uint8_t, ITC_AddressBase+0x05)

  Interrupt priority register 6/8.
- #define _ITC_SPR7 _SFR(uint8_t, ITC_AddressBase+0x06)

  Interrupt priority register 7/8.
- #define _ITC_SPR8 _SFR(uint8_t, ITC_AddressBase+0x07)

  Interrupt priority register 8/8.
- #define _ITC_SPR1_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 1/8 reset value.
- #define _ITC_SPR2_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 2/8 reset value.
- #define _ITC_SPR3_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 3/8 reset value.
- #define _ITC_SPR4_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 4/8 reset value.
- #define _ITC_SPR5_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 5/8 reset value.
- #define _ITC_SPR6_RESET_VALUE ((uint8_t) 0xFF)

Interrupt priority register 6/8 reset value.

- #define _ITC_SPR7_RESET_VALUE ((uint8_t) 0xFF)

  Interrupt priority register 7/8 reset value.
- #define _ITC_SPR8_RESET_VALUE ((uint8_t) 0x0F)

  Interrupt priority register 8/8 reset value.
- #define _ITC_VECT1SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 1 [1:0] (in _ITC_SPR1)
- #define _ITC_VECT1SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 1 [0] (in _ITC_SPR1)
- #define _ITC_VECT1SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 1 [1] (in _ITC_SPR1)
- #define _ITC_VECT4SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 4 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT4SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 4 [0] (in _ITC_SPR2)
- #define _ITC_VECT4SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 4 [1] (in _ITC_SPR2)
- #define _ITC_VECT6SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 6 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT6SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 6 [0] (in _ITC_SPR2)
- #define _ITC_VECT6SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 6 [1] (in _ITC_SPR2)
- #define _ITC_VECT7SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 7 [1:0] (in _ITC_SPR2)
- #define _ITC_VECT7SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 7 [0] (in _ITC_SPR2)
- #define _ITC_VECT7SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 7 [1] (in _ITC_SPR2)
- #define _ITC_VECT8SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 8 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT8SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 8 [0] (in _ITC_SPR3)
- #define _ITC_VECT8SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 8 [1] (in _ITC_SPR3)
- #define _ITC_VECT9SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 9 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT9SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 9 [0] (in _ITC_SPR3)
- #define _ITC_VECT9SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 9 [1] (in _ITC_SPR3)
- #define _ITC_VECT10SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 10 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT10SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 10 [0] (in _ITC_SPR3)
- #define _ITC_VECT10SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 10 [1] (in _ITC_SPR3)
- #define _ITC_VECT11SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 11 [1:0] (in _ITC_SPR3)
- #define _ITC_VECT11SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 11 [0] (in _ITC_SPR3)

- #define _ITC_VECT11SPR1 ((uint8_t) (0x01 << 7))
  ITC interrupt priority vector 11 [1] (in _ITC_SPR3)
- #define _ITC_VECT12SPR ((uint8_t) (0x03 << 0))
  ITC interrupt priority vector 12 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT12SPR0 ((uint8_t) (0x01 << 0))
  ITC interrupt priority vector 12 [0] (in _ITC_SPR4)
- #define _ITC_VECT12SPR1 ((uint8_t) (0x01 << 1))
  ITC interrupt priority vector 12 [1] (in _ITC_SPR4)
- #define _ITC_VECT13SPR ((uint8_t) (0x03 << 2))
  ITC interrupt priority vector 13 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT13SPR0 ((uint8_t) (0x01 << 2))
  ITC interrupt priority vector 13 [0] (in _ITC_SPR4)
- #define _ITC_VECT13SPR1 ((uint8_t) (0x01 << 3))
  ITC interrupt priority vector 13 [1] (in _ITC_SPR4)
- #define _ITC_VECT14SPR ((uint8_t) (0x03 << 4))
  ITC interrupt priority vector 14 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT14SPR0 ((uint8_t) (0x01 << 4))
  ITC interrupt priority vector 14 [0] (in _ITC_SPR4)
- #define _ITC_VECT14SPR1 ((uint8_t) (0x01 << 5))
  ITC interrupt priority vector 14 [1] (in _ITC_SPR4)
- #define _ITC_VECT15SPR ((uint8_t) (0x03 << 6))
  ITC interrupt priority vector 15 [1:0] (in _ITC_SPR4)
- #define _ITC_VECT15SPR0 ((uint8_t) (0x01 << 6))
  ITC interrupt priority vector 15 [0] (in _ITC_SPR4)
- #define _ITC_VECT15SPR1 ((uint8_t) (0x01 << 7))
  ITC interrupt priority vector 15 [1] (in _ITC_SPR4)
- #define _ITC_VECT19SPR ((uint8_t) (0x03 << 6))
  ITC interrupt priority vector 19 [1:0] (in _ITC_SPR5)
- #define _ITC_VECT19SPR0 ((uint8_t) (0x01 << 6))
  ITC interrupt priority vector 19 [0] (in _ITC_SPR5)
- #define _ITC_VECT19SPR1 ((uint8_t) (0x01 << 7))
  ITC interrupt priority vector 19 [1] (in _ITC_SPR5)
- #define _ITC_VECT20SPR ((uint8_t) (0x03 << 0))
  ITC interrupt priority vector 20 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT20SPR0 ((uint8_t) (0x01 << 0))
  ITC interrupt priority vector 20 [0] (in _ITC_SPR6)
- #define _ITC_VECT20SPR1 ((uint8_t) (0x01 << 1))
  ITC interrupt priority vector 20 [1] (in _ITC_SPR6)
- #define _ITC_VECT21SPR ((uint8_t) (0x03 << 2))
  ITC interrupt priority vector 21 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT21SPR0 ((uint8_t) (0x01 << 2))
  ITC interrupt priority vector 21 [0] (in _ITC_SPR6)
- #define _ITC_VECT21SPR1 ((uint8_t) (0x01 << 3))
  ITC interrupt priority vector 21 [1] (in _ITC_SPR6)
- #define _ITC_VECT22SPR ((uint8_t) (0x03 << 4))
  ITC interrupt priority vector 22 [1:0] (in _ITC_SPR6)
- #define _ITC_VECT22SPR0 ((uint8_t) (0x01 << 4))
  ITC interrupt priority vector 22 [0] (in _ITC_SPR6)
- #define _ITC_VECT22SPR1 ((uint8_t) (0x01 << 5))
  ITC interrupt priority vector 22 [1] (in _ITC_SPR6)
- #define _ITC_VECT25SPR ((uint8_t) (0x03 << 2))

ITC interrupt priority vector 25 [1:0] (in _ITC_SPR7)

- #define _ITC_VECT25SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 25 [0] (in _ITC_SPR7)
- #define _ITC_VECT25SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 25 [1] (in _ITC_SPR7)
- #define _ITC_VECT26SPR ((uint8_t) (0x03 << 4))

  ITC interrupt priority vector 26 [1:0] (in _ITC_SPR7)
- #define _ITC_VECT26SPR0 ((uint8_t) (0x01 << 4))

  ITC interrupt priority vector 26 [0] (in _ITC_SPR7)
- #define _ITC_VECT26SPR1 ((uint8_t) (0x01 << 5))

  ITC interrupt priority vector 26 [1] (in _ITC_SPR7)
- #define _ITC_VECT27SPR ((uint8_t) (0x03 << 6))

  ITC interrupt priority vector 27 [1:0] (in _ITC_SPR7)
- #define _ITC_VECT27SPR0 ((uint8_t) (0x01 << 6))

  ITC interrupt priority vector 27 [0] (in _ITC_SPR7)
- #define _ITC_VECT27SPR1 ((uint8_t) (0x01 << 7))

  ITC interrupt priority vector 27 [1] (in _ITC_SPR7)
- #define _ITC_VECT28SPR ((uint8_t) (0x03 << 0))

  ITC interrupt priority vector 28 [1:0] (in _ITC_SPR8)
- #define _ITC_VECT28SPR0 ((uint8_t) (0x01 << 0))

  ITC interrupt priority vector 28 [0] (in _ITC_SPR8)
- #define _ITC_VECT28SPR1 ((uint8_t) (0x01 << 1))

  ITC interrupt priority vector 28 [1] (in _ITC_SPR8)
- #define _ITC_VECT29SPR ((uint8_t) (0x03 << 2))

  ITC interrupt priority vector 29 [1:0] (in _ITC_SPR8)
- #define _ITC_VECT29SPR0 ((uint8_t) (0x01 << 2))

  ITC interrupt priority vector 29 [0] (in _ITC_SPR8)
- #define _ITC_VECT29SPR1 ((uint8_t) (0x01 << 3))

  ITC interrupt priority vector 29 [1] (in _ITC_SPR8)

Index