Micromouse2019

Generated by Doxygen 1.8.13

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Module Documentation

3.1 Template_Project

Functions

- void NMI_Handler (void)

  This function handles NMI exception.
- void HardFault_Handler (void)

  This function handles Hard Fault exception.
- void MemManage_Handler (void)

  This function handles Memory Manage exception.
- void BusFault_Handler (void)

  This function handles Bus Fault exception.
- void UsageFault_Handler (void)

  This function handles Usage Fault exception.
- void SVC_Handler (void)

  This function handles SVCall exception.
- void DebugMon_Handler (void)

  This function handles Debug Monitor exception.
- void PendSV_Handler (void)

  This function handles PendSVC exception.
- void SysTick_Handler (void)

  This function handles SysTick Handler.

3.1.1 Detailed Description

3.1.2 Function Documentation

3.1.2.1 BusFault_Handler()

```
void BusFault_Handler (
             void  )
```

This function handles Bus Fault exception.

Parameters

None	

Return values

None	

3.1.2.2 DebugMon_Handler()

```
void DebugMon_Handler (
             void  )
```

This function handles Debug Monitor exception.

Parameters

None	

Return values

None	

3.1.2.3 HardFault_Handler()

```
void HardFault_Handler (
             void  )
```

This function handles Hard Fault exception.

Parameters

None	

Return values

None	

3.1.2.4 MemManage_Handler()

```
void MemManage_Handler (
```

```
            void  )
```

This function handles Memory Manage exception.

Parameters

None	

Return values

None	

3.1.2.5 NMI_Handler()

```
void NMI_Handler (
            void  )
```

This function handles NMI exception.

Parameters

None	

Return values

None	

3.1.2.6 PendSV_Handler()

```
void PendSV_Handler (
            void  )
```

This function handles PendSVC exception.

Parameters

None	

Return values

None	

3.1.2.7 SVC_Handler()

```
void SVC_Handler (
             void  )
```

This function handles SVCall exception.

Parameters

None	

Return values

None	

3.1.2.8 SysTick_Handler()

```
void SysTick_Handler (
             void  )
```

This function handles SysTick Handler.

Parameters

None	

Return values

None	

3.1.2.9 UsageFault_Handler()

```
void UsageFault_Handler (
             void  )
```

This function handles Usage Fault exception.

Parameters

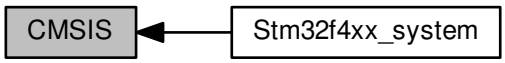
None	

Return values

None	

3.2 CMSIS

Collaboration diagram for CMSIS:



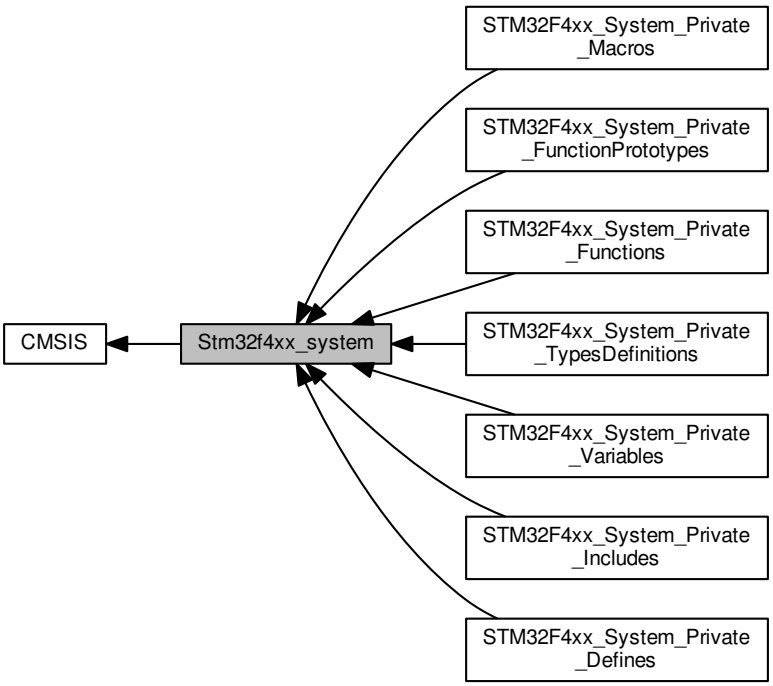
Modules

- Stm32f4xx_system

3.2.1 Detailed Description

3.3 Stm32f4xx_system

Collaboration diagram for Stm32f4xx_system:



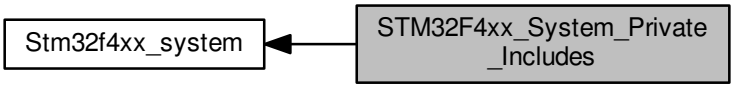
Modules

- STM32F4xx_System_Private_Includes
- STM32F4xx_System_Private_TypesDefinitions
- STM32F4xx_System_Private_Defines
- STM32F4xx_System_Private_Macros
- STM32F4xx_System_Private_Variables
- STM32F4xx_System_Private_FunctionPrototypes
- STM32F4xx_System_Private_Functions

3.3.1 Detailed Description

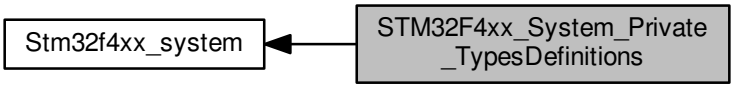
3.4 STM32F4xx_System_Private_Includes

Collaboration diagram for STM32F4xx_System_Private_Includes:



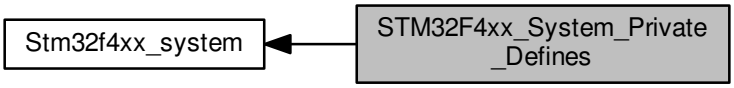
3.5 STM32F4xx_System_Private_TypesDefinitions

Collaboration diagram for STM32F4xx_System_Private_TypesDefinitions:



3.6 STM32F4xx_System_Private_Defines

Collaboration diagram for STM32F4xx_System_Private_Defines:



Macros

- #define VECT_TAB_OFFSET 0x00
- #define **PLL_M** 16 /∗ 25 ∗/
- #define **PLL_N** 336
- #define **PLL_P** 2
- #define **PLL_Q** 7

3.6.1 Detailed Description

3.6.2 Macro Definition Documentation

3.6.2.1 VECT_TAB_OFFSET

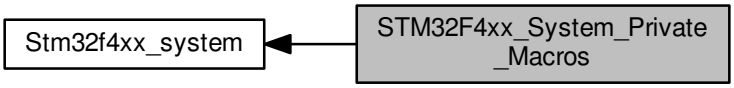
```
#define VECT_TAB_OFFSET 0x00
```

< Uncomment the following line if you need to use external SRAM mounted on STM324xG_EVAL/STM324x7I_↩ EVAL boards as data memory

< Uncomment the following line if you need to relocate your vector Table in Internal SRAM. Vector Table base offset field. This value must be a multiple of 0x200.

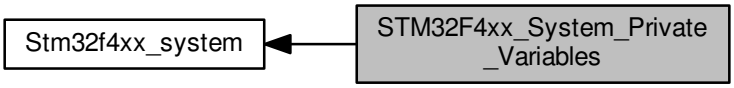
3.7 STM32F4xx_System_Private_Macros

Collaboration diagram for STM32F4xx_System_Private_Macros:



3.8 STM32F4xx_System_Private_Variables

Collaboration diagram for STM32F4xx_System_Private_Variables:



Variables

- uint32_t **SystemCoreClock** = 168000000
- __I uint8_t **AHBPrescTable** [16]

3.8.1 Detailed Description

3.8.2 Variable Documentation

3.8.2.1 AHBPrescTable

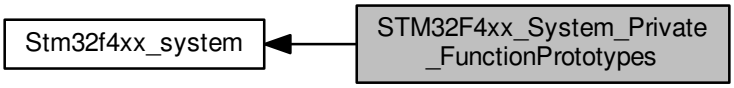
```
__I uint8_t AHBPrescTable[16]
```

Initial value:

```
=
        { 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9 }
```

3.9 STM32F4xx_System_Private_FunctionPrototypes

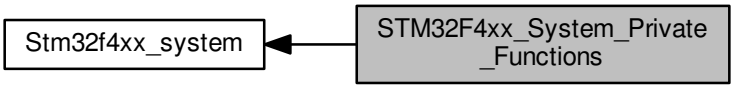
Collaboration diagram for STM32F4xx_System_Private_FunctionPrototypes:



3.9.1 Detailed Description

3.10 STM32F4xx_System_Private_Functions

Collaboration diagram for STM32F4xx_System_Private_Functions:



Functions

- void SystemInit (void)

  Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.
- void SystemCoreClockUpdate (void)

  Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

3.10.1 Detailed Description

3.10.2 Function Documentation

3.10.2.1 SystemCoreClockUpdate()

```
void SystemCoreClockUpdate (
            void  )
```

Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

Note

Each time the core clock (HCLK) changes, this function must be called to update SystemCoreClock variable value. Otherwise, any configuration based on this variable will be incorrect.
- The system frequency computed by this function is not the real frequency in the chip. It is calculated based on the predefined constant and the selected clock source:

- If SYSCLK source is HSI, SystemCoreClock will contain the HSI_VALUE(∗)
- If SYSCLK source is HSE, SystemCoreClock will contain the HSE_VALUE(∗∗)
- If SYSCLK source is PLL, SystemCoreClock will contain the HSE_VALUE(∗∗) or HSI_VALUE(∗) multiplied/divided by the PLL factors.

(∗) HSI_VALUE is a constant defined in stm32f4xx.h file (default value 16 MHz) but the real value may vary depending on the variations in voltage and temperature.

(∗∗) HSE_VALUE is a constant defined in stm32f4xx.h file (default value 25 MHz), user has to ensure that HSE_↩ VALUE is same as the real frequency of the crystal used. Otherwise, this function may have wrong result.

- The result of this function could be not correct when using fractional value for HSE crystal.

Parameters

None	

Return values

None	

3.10.2.2 SystemInit()

```
void SystemInit (
             void  )
```

Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.

Parameters

None	

Return values

None	

Chapter 4

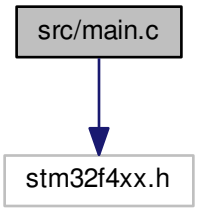
File Documentation

4.1 src/main.c File Reference

main program

```
#include "stm32f4xx.h"
```

Include dependency graph for main.c:



Functions

- int function (int a, int b)
- int main (void)

4.1.1 Detailed Description

main program

Author

Daniel Tar, Zoltar Resi, Andras Lauko

Version

V0.1

Date

04-03-2019

Attention

4.1.2 Function Documentation

4.1.2.1 function()

```
int function (
            int a,
            int b )
```

Parameters

a	- interger 1
b	- interger 2

Returns

the sum of intergers

4.1.2.2 main()

```
int main (
              void  )
```

==========

Abstract: main program

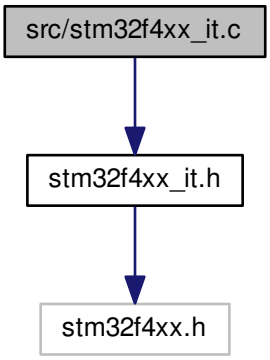
IMPORTANT NOTE! The symbol VECT_TAB_SRAM needs to be defined when building the project if code has been located to RAM and interrupts are used. Otherwise the interrupt table located in flash will be used. See also the <system_*.c> file and how the SystemInit() function updates SCB->VTOR register. E.g. SCB->VTOR = 0x20000000;

4.2 src/stm32f4xx_it.c File Reference

Main Interrupt Service Routines. This file provides template for all exceptions handler and peripherals interrupt service routine.

```
#include "stm32f4xx_it.h"
```

Include dependency graph for stm32f4xx_it.c:



Functions

- void NMI_Handler (void)

  This function handles NMI exception.
- void HardFault_Handler (void)

  This function handles Hard Fault exception.
- void MemManage_Handler (void)

  This function handles Memory Manage exception.
- void BusFault_Handler (void)

  This function handles Bus Fault exception.
- void UsageFault_Handler (void)

This function handles Usage Fault exception.

- void SVC_Handler (void)

  This function handles SVCall exception.

- void DebugMon_Handler (void)

  This function handles Debug Monitor exception.

- void PendSV_Handler (void)

  This function handles PendSVC exception.

- void SysTick_Handler (void)

  This function handles SysTick Handler.

4.2.1 Detailed Description

Main Interrupt Service Routines. This file provides template for all exceptions handler and peripherals interrupt service routine.

Author

MCD Application Team

Version

V1.1.0

Date

18-January-2013

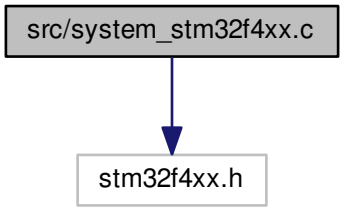
Attention

4.3 src/system_stm32f4xx.c File Reference

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File. This file contains the system clock configuration for STM32F4xx devices, and is generated by the clock configuration tool stm32f4xx_Clock_Configuration↩_V1.1.0.xls.

```
#include "stm32f4xx.h"
```

Include dependency graph for system_stm32f4xx.c:



Macros

- #define VECT_TAB_OFFSET 0x00
- #define **PLL_M** 16 /∗ 25 ∗/
- #define **PLL_N** 336
- #define **PLL_P** 2
- #define **PLL_Q** 7

Functions

- void SystemInit (void)

  Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.
- void SystemCoreClockUpdate (void)

  Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

Variables

- uint32_t **SystemCoreClock** = 168000000
- __I uint8_t **AHBPrescTable** [16]

4.3.1 Detailed Description

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File. This file contains the system clock configuration for STM32F4xx devices, and is generated by the clock configuration tool stm32f4xx_Clock_Configuration↩_V1.1.0.xls.

Author

MCD Application Team

Version

V1.1.0

Date

18-January-2013

1. This file provides two functions and one global variable to be called from user application:
   - SystemInit(): Setups the system clock (System clock source, PLL Multiplier and Divider factors, A↩HB/APBx prescalers and Flash settings), depending on the configuration made in the clock xls tool. This function is called at startup just after reset and before branch to main program. This call is made inside the "startup_stm32f4xx.s" file.
   - SystemCoreClock variable: Contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.
   - SystemCoreClockUpdate(): Updates the variable SystemCoreClock and must be called whenever the core clock is changed during program execution.
2. After each device reset the HSI (16 MHz) is used as system clock source. Then SystemInit() function is called, in "startup_stm32f4xx.s" file, to configure the system clock before to branch to main program.
3. If the system clock source selected by user fails to startup, the SystemInit() function will do nothing and HSI still used as system clock source. User can add some code to deal with this issue inside the SetSysClock() function.
4. The default value of HSE crystal is set to 25MHz, refer to "HSE_VALUE" define in "stm32f4xx.h" file. When HSE is used as system clock source, directly or through PLL, and you are using different crystal you have to adapt the HSE value to your own configuration.

5. This file configures the system clock as follows:

==========

Supported STM32F40xx/41xx/427x/437x devices

System Clock source | PLL (HSI) // (HSE)

SYSCLK(Hz) | 168000000

HCLK(Hz) | 168000000

AHB Prescaler | 1

APB1 Prescaler | **4**

APB2 Prescaler | **2**

HSI Frequency (Hz) // HSE Frequency(Hz) | **16000000 // 25000000**

PLL_M | **16 // 25**

PLL_N | **336**

PLL_P | **2**

PLL_Q | **7**

PLLI2S_N | **NA**

PLLI2S_R | **NA**

I2S input clock | **NA**

VDD(V) | **3.3**

Main regulator output voltage | **Scale1 mode**

Flash Latency(WS) | **5**

Prefetch Buffer | **ON**

Instruction cache | **ON**

Data cache | **ON**

Require 48MHz for USB OTG FS, | Disabled

SDIO and RNG clock |

==========

Attention

Index