

CLI STM32 FreeRTOS -- By Aaron Escoboza

Generated by Doxygen 1.10.0

1 Topic Index	1
1.1 Topics	1
2 Data Structure Index	3
2.1 Data Structures	3
3 File Index	5
3.1 File List	5
4 Topic Documentation	7
4.1 CMSIS	7
4.1.1 Detailed Description	7
4.1.2 Stm32f4xx_system	7
4.1.2.1 Detailed Description	7
4.1.2.2 STM32F4xx_System_Private_Includes	7
4.1.2.3 STM32F4xx_System_Private_TypesDefinitions	8
4.1.2.4 STM32F4xx_System_Private_Defines	8
4.1.2.5 STM32F4xx_System_Private_Macros	8
4.1.2.6 STM32F4xx_System_Private_Variables	8
4.1.2.7 STM32F4xx_System_Private_FunctionPrototypes	8
4.1.2.8 STM32F4xx_System_Private_Functions	8
5 Data Structure Documentation	13
5.1 Pwm_TIM_OC_InitTypeDef Struct Reference	13
5.2 PwmConfigStruct Struct Reference	13
6 File Documentation	15
6.1 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/inc/bsp.h File Reference	15
6.1.1 Detailed Description	15
6.1.2 Function Documentation	15
6.1.2.1 bsplnit()	15
6.2 bsp.h	16
6.3 bspClk.h	16
6.4 bspGpio.h	16
6.5 bspPwm.h	17
6.6 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/src/bsp.c File Reference	17
6.6.1 Detailed Description	18
6.6.2 Function Documentation	18
6.6.2.1 bsplnit()	18
6.6.2.2 clkInit()	18
6.6.2.3 consoleInit()	19
6.6.2.4 heartBeatInit()	19
6.7 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/src/bspClk.c File Reference	19

6.7.1 Detailed Description	20
6.7.2 Function Documentation	20
6.7.2.1 bspGetClockInfo()	20
6.8 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/src/bspGpio.c File Reference	20
6.8.1 Detailed Description	21
6.8.2 Function Documentation	21
6.8.2.1 bspGpioMapInstance()	21
6.8.2.2 bspGpioRead()	21
6.8.2.3 bspGpioToggle()	21
6.8.2.4 bspGpioWrite()	22
6.8.2.5 bspMapInstanceToHal()	22
6.8.2.6 bspMapPinNumFromBspToHal()	23
6.9 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Inc/appConfig.h File Reference	23
6.9.1 Detailed Description	23
6.10 appConfig.h	24
6.11 console.h	24
6.12 FreeRTOSConfig.h	24
6.13 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Inc/main.h File Reference	26
6.13.1 Detailed Description	26
6.13.2 Function Documentation	26
6.13.2.1 Error_Handler()	26
6.14 main.h	27
6.15 stm32f4xx_hal_conf.h	27
6.16 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Inc/stm32f4xx_it.h File Reference	32
6.16.1 Detailed Description	33
6.17 stm32f4xx_it.h	33
6.18 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/console.c File Reference	34
6.18.1 Detailed Description	36
6.18.2 Function Documentation	36
6.18.2.1 HAL_UART_RxCpltCallback()	36
6.18.2.2 prvCommandClk()	36
6.18.2.3 prvCommandEcho()	37
6.18.2.4 prvCommandGpioRead()	37
6.18.2.5 prvCommandGpioWrite()	37
6.18.2.6 prvCommandHeap()	38
6.18.2.7 prvCommandPwmSetDuty()	38
6.18.2.8 prvCommandPwmSetFreq()	39
6.18.2.9 prvCommandTaskStats()	39
6.18.2.10 prvCommandTicks()	39

6.18.2.11 prvpcMapTaskState()	40
6.18.2.12 vConsoleEnableRxInterrupt()	40
6.18.2.13 vConsoleWrite()	40
6.18.2.14 vTaskConsole()	41
6.18.2.15 xConsoleInit()	41
6.18.2.16 xConsoleRead()	41
6.18.3 Variable Documentation	42
6.18.3.1 prvpcTaskListHeader	42
6.19 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/main.c File Reference	42
6.19.1 Detailed Description	43
6.19.2 Function Documentation	43
6.19.2.1 Error_Handler()	43
6.19.2.2 HAL_TIM_PeriodElapsedCallback()	43
6.20 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/stm32f4xx_hal_timebase_tim.c File Reference	43
6.20.1 Detailed Description	44
6.20.2 Function Documentation	44
6.20.2.1 HAL_InitTick()	44
6.20.2.2 HAL_ResumeTick()	45
6.20.2.3 HAL_SuspendTick()	45
6.21 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/syscalls.c File Reference	46
6.21.1 Detailed Description	46
6.22 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/systemem.c File Reference	47
6.22.1 Detailed Description	47
6.22.2 Function Documentation	47
6.22.2.1 _sbrk()	47
6.22.3 Variable Documentation	48
6.22.3.1 __sbrk_heap_end	48
6.23 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/system_stm32f4xx.c File Reference	48
6.23.1 Detailed Description	49

Chapter 1

Topic Index

1.1 Topics

Here is a list of all topics with brief descriptions:

CMSIS	7
Stm32f4xx_system	7
STM32F4xx_System_Private_Includes	7
STM32F4xx_System_Private_TypesDefinitions	8
STM32F4xx_System_Private_Defines	8
STM32F4xx_System_Private_Macros	8
STM32F4xx_System_Private_Variables	8
STM32F4xx_System_Private_FunctionPrototypes	8
STM32F4xx_System_Private_Functions	8

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Pwm_TIM_OC_InitTypeDef	13
PwmConfigStruct	13

Chapter 3

File Index

3.1 File List

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

C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/inc/bsp.h	
Header file to expose initialization functions	15
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/inc/bspClk.h	16
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/inc/bspGpio.h	16
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/inc/bspPwm.h	17
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/src/bsp.c	
Source file to implement low level initializations	17
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/src/bspClk.c	
Source file to implement functions related to the clock tree	19
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/src/bspGpio.c	
Source file to implement functions related to GPIO operations	20
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Inc/appConfig.h	
Hold general application configuration	23
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Inc/console.h	24
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Inc/FreeRTOSConfig.h	24
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Inc/main.h	
: Header for main.c file. This file contains the common defines of the application	26
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Inc/stm32f4xx_hal_conf.h	27
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Inc/stm32f4xx_it.h	
This file contains the headers of the interrupt handlers	32
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/console.c	
Command Line Interpreter based on FreeRTOS and STM32 HAL layer Github account: https://github.com/aaron-ev	34
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/main.c	
Command Line Interpreter based on FreeRTOS and STM32 HAL layer Github account: https://github.com/aaron-ev	42
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/stm32f4xx_hal_timebase_tim.c	
HAL time base based on the hardware TIM	43
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/syscalls.c	
STM32CubeIDE Minimal System calls file	46
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/system.c	
STM32CubeIDE System Memory calls file	47
C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/system_stm32f4xx.c	
CMSIS Cortex-M4 Device Peripheral Access Layer System Source File	48

Chapter 4

Topic Documentation

4.1 CMSIS

Topics

- [Stm32f4xx_system](#)

4.1.1 Detailed Description

4.1.2 Stm32f4xx_system

Topics

- [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](#)

4.1.2.1 Detailed Description

4.1.2.2 STM32F4xx_System_Private_Includes

Macros

- `#define HSE_VALUE ((uint32_t)25000000)`
- `#define HSI_VALUE ((uint32_t)16000000)`

4.1.2.2.1 Detailed Description

4.1.2.2.2 Macro Definition Documentation

4.1.2.2.2.1 HSE_VALUE

```
#define HSE_VALUE ((uint32_t)25000000)
```

Default value of the External oscillator in Hz

4.1.2.2.2.2 HSI_VALUE

```
#define HSI_VALUE ((uint32_t)16000000)
```

Value of the Internal oscillator in Hz

4.1.2.3 STM32F4xx_System_Private_TypesDefinitions

4.1.2.4 STM32F4xx_System_Private_Defines

4.1.2.5 STM32F4xx_System_Private_Macros

4.1.2.6 STM32F4xx_System_Private_Variables

Variables

- uint32_t **SystemCoreClock** = 16000000
- const uint8_t **AHBPrescTable** [16] = {0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9}
- const uint8_t **APBPrescTable** [8] = {0, 0, 0, 0, 1, 2, 3, 4}

4.1.2.6.1 Detailed Description

4.1.2.7 STM32F4xx_System_Private_FunctionPrototypes

4.1.2.8 STM32F4xx_System_Private_Functions

Functions

- void [SystemInit](#) (void)
Setup the microcontroller system Initialize the FPU setting, vector table location and External memory configuration.
- 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.

4.1.2.8.1 Detailed Description

4.1.2.8.2 Function Documentation

4.1.2.8.2.1 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_hal_conf.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_hal_conf.h](#) file (its value depends on the application requirements), 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	
------	--

4.1.2.8.2.2 SystemInit()

```
void SystemInit (  
    void )
```

Setup the microcontroller system Initialize the FPU setting, vector table location and External memory configuration.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

Chapter 5

Data Structure Documentation

5.1 Pwm_TIM_OC_InitTypeDef Struct Reference

Data Fields

- TIM_OC_InitTypeDef **xOclnit**
- uint16_t **channel**
- uint8_t **uDuty**

The documentation for this struct was generated from the following file:

- C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/src/bspPwm.c

5.2 PwmConfigStruct Struct Reference

Data Fields

- TIM_HandleTypeDef **xTimHandle**
- [Pwm_TIM_OC_InitTypeDef](#) **uChannelXConfig** [PWM_MAX_CHANNELS]

The documentation for this struct was generated from the following file:

- C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/src/bspPwm.c

Chapter 6

File Documentation

6.1 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/inc/bsp.h File Reference

Header file to expose initialization functions.

```
#include "appConfig.h"
#include "stm32f4xx.h"
#include "stm32f4xx_hal.h"
#include "bspPwm.h"
#include "bspGpio.h"
#include "bspClk.h"
#include "errno.h"
```

Enumerations

- enum **BspError_e** { **BSP_NO_ERROR** , **BSP_ERROR_EIO** = EIO }

Functions

- BspError_e **bsplnit** (void)
Calls all BSP init functions.

6.1.1 Detailed Description

Header file to expose initialization functions.

Author

Aaron Escoboza

6.1.2 Function Documentation

6.1.2.1 bsplnit()

```
BspError_e bsplnit (
    void )
```

Calls all BSP init functions.

Parameters

<i>void</i>	
-------------	--

Return values

<i>BspError</i> ↵ _e	
-------------------------	--

6.2 bsp.h

[Go to the documentation of this file.](#)

```

00001
00010 #ifndef __BSP__H
00011 #define __BSP__H
00012
00013 #include "appConfig.h"
00014 #include "stm32f4xx.h"
00015 #include "stm32f4xx_hal.h"
00016 #include "bspPwm.h"
00017 #include "bspGpio.h"
00018 #include "bspClk.h"
00019 #include "errno.h"
00020
00021 typedef enum
00022 {
00023     BSP_NO_ERROR,
00024     BSP_ERROR_EIO = EIO,
00025 } BspError_e;
00026
00027 BspError_e bspInit(void);
00028
00029 #endif

```

6.3 bspClk.h

```

00001 #ifndef __BSP_CLK_H
00002 #define __BSP_CLK_H
00003
00004 #include "stdint.h"
00005 #include "stm32f4xx_hal.h"
00006
00007 void bspGetClockInfo(char *pcWriteBuffer, size_t xWriteBufferLen);
00008
00009 #endif

```

6.4 bspGpio.h

```

00001 #ifndef __BSP_GPIO_H
00002 #define __BSP_GPIO_H
00003
00004 #include "stdint.h"
00005 #include "stm32f4xx_hal.h"
00006
00007 typedef enum
00008 {
00009     BSP_GPIOA,
00010     BSP_GPIOB,
00011     BSP_GPIOC,
00012     BSP_GPIOD,
00013     BSP_GPIOE,
00014     BSP_GPIOH,
00015     BSP_MAX_GPIO_INSTANCE,
00016 }BspGpioInstance_e;
00017
00018 typedef enum

```

```

00019 {
00020     BSP_GPIO_PIN_0,
00021     BSP_GPIO_PIN_1,
00022     BSP_GPIO_PIN_2,
00023     BSP_GPIO_PIN_3,
00024     BSP_GPIO_PIN_4,
00025     BSP_GPIO_PIN_5,
00026     BSP_GPIO_PIN_6,
00027     BSP_GPIO_PIN_7,
00028     BSP_GPIO_PIN_8,
00029     BSP_GPIO_PIN_9,
00030     BSP_GPIO_PIN_10,
00031     BSP_GPIO_PIN_11,
00032     BSP_GPIO_PIN_12,
00033     BSP_GPIO_PIN_13,
00034     BSP_GPIO_PIN_14,
00035     BSP_GPIO_PIN_15,
00036 }BspPinNum_e;
00037
00038 typedef enum
00039 {
00040     BSP_GPIO_PIN_LOW,
00041     BSP_GPIO_PIN_HIGH,
00042 }BspGpioPinState_e;
00043
00044 void bspGpioToggle(BspGpioInstance_e eGpio, BspPinNum_e pinNum);
00045 BspGpioInstance_e bspGpioMapInstance(const char pcGpioInstance);
00046 BspGpioPinState_e bspGpioRead(BspGpioInstance_e eGpio, BspPinNum_e pinNum);
00047 void bspGpioWrite(BspGpioInstance_e eGpio, BspPinNum_e pinNum, BspGpioPinState_e pinState);
00048
00049 #endif

```

6.5 bspPwm.h

```

00001 #ifndef __BSP_PWM_H
00002 #define __BSP_PWM_H
00003
00004 #include "stdint.h"
00005 #include "stm32f4xx_hal.h"
00006
00007 typedef enum
00008 {
00009     PWM_CH_1,
00010     PWM_CH_2,
00011     PWM_CH_3,
00012     PWM_CH_4,
00013     MAX_PWM_CH,
00014 } pwmChannels_e;
00015
00016 HAL_StatusTypeDef bspPwmSetFreq(uint32_t uNewFreq);
00017 TIM_HandleTypeDef* bspPwmGetHandler(void);
00018 void bspPwmStart(pwmChannels_e ePwmChannel);
00019 HAL_StatusTypeDef bspPwmSetDuty(uint8_t uNewDuty, pwmChannels_e xChannel);
00020 HAL_StatusTypeDef bspPwmInit(void);
00021
00022 #endif

```

6.6 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/src/bsp.c File Reference

source file to implement low level initializations.

```
#include "bsp.h"
```

Functions

- static BspError_e [clkInit](#) (void)
Initialize system clocks, PLL and Clock dividers.

- static void `heartBeatInit` (void)
Initialize GPIO pin for heart beat functionality.
- BspError_e `consoleInit` (void)
Initialize UART frame.
- BspError_e `bspInit` (void)
Calls all BSP init functions.

Variables

- UART_HandleTypeDef `consoleHandle`

6.6.1 Detailed Description

source file to implement low level initializations.

Author

Aaron Escoboza

6.6.2 Function Documentation

6.6.2.1 bspInit()

```
BspError_e bspInit (
    void )
```

Calls all BSP init functions.

Parameters

<i>void</i>	
-------------	--

Return values

<i>BspError_e</i>	
-------------------	--

6.6.2.2 clkInit()

```
static BspError_e clkInit (
    void ) [static]
```

Initialize system clocks, PLL and Clock dividers.

Parameters

<i>void</i>	
-------------	--

Return values

<i>BspError</i> ↔ _e	
-------------------------	--

6.6.2.3 consoleInit()

```
BspError_e consoleInit (  
    void )
```

Initialize UART frame.

Parameters

<i>void</i>	
-------------	--

Return values

<i>BSP</i>	error
------------	-------

6.6.2.4 heartBeatInit()

```
static void heartBeatInit (  
    void ) [static]
```

Initialize GPIO pin for heart beat functionality.

Parameters

<i>void</i>	
-------------	--

Return values

<i>void</i>	
-------------	--

6.7 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/src/bspClk.c File Reference

source file to implement functions related to the clock tree.

```
#include "bspClk.h"
```

Functions

- void [bspGetClockInfo](#) (char *pcWriteBuffer, size_t xWriteBufferLen)
Gets system clock, PCLKx and CLK dividers.

6.7.1 Detailed Description

source file to implement functions related to the clock tree.

Author

Aaron Escoboza

6.7.2 Function Documentation

6.7.2.1 bspGetClockInfo()

```
void bspGetClockInfo (
    char * pcWriteBuffer,
    size_t xWriteBufferLen )
```

Gets system clock, PCLKx and CLK dividers.

Parameters

<i>*pcWriteBuffer</i>	pointer to buffer where clock information will be stored.
<i>xWriteBufferLen</i>	buffer length.

Return values

<i>void</i>	
-------------	--

6.8 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/src/bspGpio.c File Reference

source file to implement functions related to GPIO operations.

```
#include "bspGpio.h"
```

Functions

- static uint16_t [bspMapPinNumFromBspToHal](#) (uint16_t uGpioNumber)
Maps a number from the range 0 - 15 to STM32 pin macro value.
- BspGpioInstance_e [bspGpioMapInstance](#) (const char pcGpioInstance)
Maps a letter (a, b, c, d, e, h) to a BSP GPIO instance.
- static GPIO_TypeDef * [bspMapInstanceToHal](#) (BspGpioInstance_e eGpio)
Maps from a BSP instance to HAL GPIO instance pointer.
- void [bspGpioToggle](#) (BspGpioInstance_e eGpio, BspPinNum_e pinNum)
Toggles a GPIO pin.
- void [bspGpioWrite](#) (BspGpioInstance_e eGpio, BspPinNum_e pinNum, BspGpioPinState_e pinState)
Writes to a GPIO pin.
- BspGpioPinState_e [bspGpioRead](#) (BspGpioInstance_e eGpio, BspPinNum_e pinNum)
Reads from a GPIO pin.

6.8.1 Detailed Description

source file to implement functions related to GPIO operations.

Author

Aaron Escoboza

6.8.2 Function Documentation

6.8.2.1 bspGpioMapInstance()

```
BspGpioInstance_e bspGpioMapInstance (
    const char pcGpioInstance )
```

Maps a letter (a, b, c, d, e , h) to a BSP GPIO instance.

Parameters

<i>pcGpioInstance</i>	GPIO instance character.
-----------------------	--------------------------

Return values

<i>BSP</i>	GPIO instance.
------------	----------------

6.8.2.2 bspGpioRead()

```
BspGpioPinState_e bspGpioRead (
    BspGpioInstance_e eGpio,
    BspPinNum_e pinNum )
```

Reads from a GPIO pin.

Parameters

<i>eGpio</i>	BSP GPIO instance.
<i>pinNum</i>	BSP GPIO pin number.

Return values

<i>BSP</i>	pin state
------------	-----------

6.8.2.3 bspGpioToggle()

```
void bspGpioToggle (
    BspGpioInstance_e eGpio,
    BspPinNum_e pinNum )
```

Toggles a GPIO pin.

Parameters

<i>eGpio</i>	BSP GPIO instance.
<i>pinNum</i>	BSP GPIO pin number.

Return values

<i>void</i>	
-------------	--

6.8.2.4 bspGpioWrite()

```
void bspGpioWrite (
    BspGpioInstance_e eGpio,
    BspPinNum_e pinNum,
    BspGpioPinState_e pinState )
```

Writes to a GPIO pin.

Parameters

<i>eGpio</i>	BSP GPIO instance.
<i>pinNum</i>	BSP GPIO pin number.
<i>pinState</i>	new BSP pin state.

Return values

<i>void</i>	
-------------	--

6.8.2.5 bspMapInstanceToHal()

```
static GPIO_TypeDef * bspMapInstanceToHal (
    BspGpioInstance_e eGpio ) [static]
```

Maps from a BSP instance to HAL GPIO instance pointer.

Parameters

<i>eGpio</i>	BSP GPIO instance.
--------------	--------------------

Return values

<i>HAL</i>	GPIO instance pointer.
------------	------------------------

6.8.2.6 bspMapPinNumFromBspToHal()

```
static uint16_t bspMapPinNumFromBspToHal (
    uint16_t uGpioNumber ) [static]
```

Maps a number from the range 0 - 15 to STM32 pin macro value.

Parameters

<i>uGpioNumber</i>	Pin number.
--------------------	-------------

Return values

<i>STM32</i>	HAL pin number.
--------------	-----------------

6.9 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Inc/appConfig.h File Reference

Hold general application configuration.

Macros

- #define **DEBUG_PRINT_EN** 1 /* 1 = Enable , 0 = Disable */
- #define **HEART_BEAT_PRIORITY_TASK** 1
- #define **HEART_BEAT_LED_PORT** GPIOC
- #define **HEART_BEAT_LED_PIN** GPIO_PIN_13
- #define **HEART_BEAT_BLINK_DELAY** 500 /* In ms */
- #define **CONSOLE_INSTANCE** USART1
- #define **CONSOLE_TX_PIN** GPIO_PIN_6
- #define **CONSOLE_RX_PIN** GPIO_PIN_7
- #define **CONSOLE_GPIO_PORT** GPIOB
- #define **CONSOLE_BAUDRATE** 9600
- #define **CONSOLE_TASK_PRIORITY** 1
- #define **CONSOLE_STACK_SIZE** 3000
- #define **PWM_GPIO_INSTANCE** GPIOA
- #define **PWM_GPIO_PINX** GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3
- #define **PWM_GPIO_ALTERNATE** GPIO_AF1_TIM2
- #define **PWM_TIM_INSTANCE** TIM2
- #define **PWM_TIM_CHANNEL** TIM_CHANNEL_1
- #define **PWM_DEFAULT_PULSE** 500

6.9.1 Detailed Description

Hold general application configuration.

Author

Aaron Escoboza

6.10 appConfig.h

[Go to the documentation of this file.](#)

```

00001
00009 #ifndef APP_CONFIG_H
00010 #define APP_CONFIG_H
00011
00012 /* Enable debug messages */
00013 #define DEBUG_PRINT_EN 1 /* 1 = Enable , 0 = Disable */
00014
00015 /* Task priorities */
00016 #define HEART_BEAT_PRIORITY_TASK 1
00017
00018 /* Heart beat settings */
00019 #define HEART_BEAT_LED_PORT GPIOC
00020 #define HEART_BEAT_LED_PIN GPIO_PIN_13
00021 #define HEART_BEAT_BLINK_DELAY 500 /* In ms */
00022
00023 /* CLI console settings */
00024 #define CONSOLE_INSTANCE USART1
00025 #define CONSOLE_TX_PIN GPIO_PIN_6
00026 #define CONSOLE_RX_PIN GPIO_PIN_7
00027 #define CONSOLE_GPIO_PORT GPIOB
00028 #define CONSOLE_BAUDRATE 9600
00029 #define CONSOLE_TASK_PRIORITY 1
00030 #define CONSOLE_STACK_SIZE 3000
00031
00032 /* PWM signal settings */
00033 #define PWM_GPIO_INSTANCE GPIOA
00034 #define PWM_GPIO_PINX GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3
00035 #define PWM_GPIO_ALTERNATE GPIO_AF1_TIM2
00036 #define PWM_TIM_INSTANCE TIM2
00037 #define PWM_TIM_CHANNEL TIM_CHANNEL_1
00038 #define PWM_DEFAULT_PULSE 500
00039
00040 #endif

```

6.11 console.h

```

00001
00002 #ifndef __CONSOLE_H
00003 #define __CONSOLE_H
00004
00005 #include "FreeRTOS.h"
00006
00007 BaseType_t xConsoleInit(uint16_t usStackSize, UBaseType_t uxPriority, UART_HandleTypeDef
    *pxUartHandle);
00008
00009 #endif

```

6.12 FreeRTOSConfig.h

```

00001 /*
00002  * FreeRTOS V202112.00
00003  * Copyright (C) 2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
00004  *
00005  * Permission is hereby granted, free of charge, to any person obtaining a copy of
00006  * this software and associated documentation files (the "Software"), to deal in
00007  * the Software without restriction, including without limitation the rights to
00008  * use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
00009  * the Software, and to permit persons to whom the Software is furnished to do so,
00010  * subject to the following conditions:
00011  *
00012  * The above copyright notice and this permission notice shall be included in all
00013  * copies or substantial portions of the Software.
00014  *
00015  * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00016  * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
00017  * FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
00018  * COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
00019  * IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
00020  * CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
00021  *
00022  * http://www.FreeRTOS.org
00023  * http://aws.amazon.com/freertos
00024  *
00025  * 1 tab == 4 spaces!

```

```

00026 */
00027
00028 #ifndef FREERTOS_CONFIG_H
00029 #define FREERTOS_CONFIG_H
00030
00031 /*-----
00032  * Application specific definitions.
00033  *
00034  * These definitions should be adjusted for your particular hardware and
00035  * application requirements.
00036  *
00037  * THESE PARAMETERS ARE DESCRIBED WITHIN THE 'CONFIGURATION' SECTION OF THE
00038  * FreeRTOS API DOCUMENTATION AVAILABLE ON THE FreeRTOS.org WEB SITE.
00039  *
00040  * See http://www.freertos.org/a00110.html
00041  *-----*/
00042
00043 /* Ensure stdint is only used by the compiler, and not the assembler. */
00044 #if defined(__ICCARM__) || defined(__GNUC__) || defined(_CC_ARM)
00045     #include <stdint.h>
00046     extern uint32_t SystemCoreClock;
00047 #endif
00048
00049 #define configUSE_PREEMPTION 1
00050 #define configUSE_IDLE_HOOK 0
00051 #define configUSE_TICK_HOOK 0
00052 #define configCPU_CLOCK_HZ (SystemCoreClock)
00053 #define configTICK_RATE_HZ ((TickType_t)1000)
00054 #define configMAX_PRIORITIES (5)
00055 #define configMINIMAL_STACK_SIZE ((unsigned short)130)
00056 // #define configTOTAL_HEAP_SIZE ((size_t)(75 * 1024))
00057 #define configTOTAL_HEAP_SIZE ((size_t)(75 * 524))
00058 #define configMAX_TASK_NAME_LEN (10)
00059 #define configUSE_TRACE_FACILITY 1
00060 #define configUSE_16_BIT_TICKS 0
00061 #define configIDLE_SHOULD_YIELD 1
00062 #define configUSE_MUTEXES 1
00063 #define configQUEUE_REGISTRY_SIZE 8
00064 #define configCHECK_FOR_STACK_OVERFLOW 0
00065 #define configUSE_RECURSIVE_MUTEXES 1
00066 #define configUSE_MALLOC_FAILED_HOOK 0
00067 #define configUSE_APPLICATION_TASK_TAG 0
00068 #define configUSE_COUNTING_SEMAPHORES 1
00069 #define configGENERATE_RUN_TIME_STATS 0
00070 #define configUSE_TASK_NOTIFICATIONS 1
00071 #define configTASK_NOTIFICATION_ARRAY_ENTRIES 2
00072 #define configCOMMAND_INT_MAX_OUTPUT_SIZE 500
00073
00074 /* Co-routine definitions. */
00075 #define configUSE_CO_ROUTINES 0
00076 #define configMAX_CO_ROUTINE_PRIORITIES (2)
00077
00078 /* Software timer definitions. */
00079 #define configUSE_TIMERS 1
00080 #define configTIMER_TASK_PRIORITY (2)
00081 #define configTIMER_QUEUE_LENGTH 10
00082 #define configTIMER_TASK_STACK_DEPTH (configMINIMAL_STACK_SIZE * 2)
00083
00084 /* Set the following definitions to 1 to include the API function, or zero
00085 to exclude the API function. */
00086 #define INCLUDE_vTaskPrioritySet 1
00087 #define INCLUDE_uxTaskPriorityGet 1
00088 #define INCLUDE_vTaskDelete 1
00089 #define INCLUDE_vTaskCleanUpResources 1
00090 #define INCLUDE_vTaskSuspend 1
00091 #define INCLUDE_vTaskDelayUntil 1
00092 #define INCLUDE_vTaskDelay 1
00093
00094 /* Cortex-M specific definitions. */
00095 #ifdef __NVIC_PRIO_BITS
00096 /* __NVIC_PRIO_BITS will be specified when CMSIS is being used. */
00097 #define configPRIO_BITS __NVIC_PRIO_BITS
00098 #else
00099 #define configPRIO_BITS 4 /* 15 priority levels */
00100 #endif
00101
00102 /* The lowest interrupt priority that can be used in a call to a "set priority"
00103 function. */
00104 #define configLIBRARY_LOWEST_INTERRUPT_PRIORITY 0xf
00105
00106 /* The highest interrupt priority that can be used by any interrupt service
00107 routine that makes calls to interrupt safe FreeRTOS API functions. DO NOT CALL
00108 INTERRUPT SAFE FREERTOS API FUNCTIONS FROM ANY INTERRUPT THAT HAS A HIGHER
00109 PRIORITY THAN THIS! (higher priorities are lower numeric values. */
00110 #define configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY 5
00111
00112 /* Interrupt priorities used by the kernel port layer itself. These are generic

```

```

00113 to all Cortex-M ports, and do not rely on any particular library functions. */
00114 #define configKERNEL_INTERRUPT_PRIORITY (configLIBRARY_LOWEST_INTERRUPT_PRIORITY « (8 -
    configPRIO_BITS))
00115 /* !!!! configMAX_SYSCALL_INTERRUPT_PRIORITY must not be set to zero !!!!
00116 See http://www.FreeRTOS.org/RTOS-Cortex-M3-M4.html. */
00117 #define configMAX_SYSCALL_INTERRUPT_PRIORITY (configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY « (8 -
    configPRIO_BITS))
00118
00119 /* Normal assert() semantics without relying on the provision of an assert.h
00120 header file. */
00121 #define configASSERT(x) \
00122     if ((x) == 0) \
00123     { \
00124         taskDISABLE_INTERRUPTS(); \
00125         for (;;) \
00126             ; \
00127     }
00128
00129 /* Definitions that map the FreeRTOS port interrupt handlers to their CMSIS
00130 standard names. */
00131 #define vPortSVCHandler SVC_Handler
00132 #define xPortPendSVHandler PendSV_Handler
00133 #define xPortSysTickHandler SysTick_Handler
00134
00135 #endif /* FREERTOS_CONFIG_H */

```

6.13 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Inc/main.h File Reference

: Header for [main.c](#) file. This file contains the common defines of the application.

```
#include "stm32f4xx_hal.h"
```

Functions

- void [Error_Handler](#) (void)
This function is executed in case of error occurrence.

6.13.1 Detailed Description

: Header for [main.c](#) file. This file contains the common defines of the application.

Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

6.13.2 Function Documentation

6.13.2.1 Error_Handler()

```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

Return values

None	
------	--

6.14 main.h

[Go to the documentation of this file.](#)

```

00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020
00021 /* Define to prevent recursive inclusion -----*/
00022 #ifndef __MAIN_H
00023 #define __MAIN_H
00024
00025 #ifdef __cplusplus
00026 extern "C" {
00027 #endif
00028
00029 /* Includes -----*/
00030 #include "stm32f4xx_hal.h"
00031
00032 /* Private includes -----*/
00033 /* USER CODE BEGIN Includes */
00034
00035 /* USER CODE END Includes */
00036
00037 /* Exported types -----*/
00038 /* USER CODE BEGIN ET */
00039
00040 /* USER CODE END ET */
00041
00042 /* Exported constants -----*/
00043 /* USER CODE BEGIN EC */
00044
00045 /* USER CODE END EC */
00046
00047 /* Exported macro -----*/
00048 /* USER CODE BEGIN EM */
00049
00050 /* USER CODE END EM */
00051
00052 /* Exported functions prototypes -----*/
00053 void Error_Handler(void);
00054
00055 /* USER CODE BEGIN EFP */
00056
00057 /* USER CODE END EFP */
00058
00059 /* Private defines -----*/
00060
00061 /* USER CODE BEGIN Private defines */
00062
00063 /* USER CODE END Private defines */
00064
00065 #ifdef __cplusplus
00066 }
00067 #endif
00068
00069 #endif /* __MAIN_H */

```

6.15 stm32f4xx_hal_conf.h

```

00001 /* USER CODE BEGIN Header */
00021 /* USER CODE END Header */
00022
00023 /* Define to prevent recursive inclusion -----*/
00024 #ifndef __STM32F4xx_HAL_CONF_H
00025 #define __STM32F4xx_HAL_CONF_H
00026
00027 #ifdef __cplusplus
00028 extern "C" {
00029 #endif
00030
00031 /* Exported types -----*/

```

```

00032 /* Exported constants -----*/
00033
00034 /* ##### Module Selection ##### */
00038 #define HAL_MODULE_ENABLED
00039
00040 /* #define HAL_CRYPT_MODULE_ENABLED */
00041 /* #define HAL_ADC_MODULE_ENABLED */
00042 /* #define HAL_CAN_MODULE_ENABLED */
00043 /* #define HAL_CRC_MODULE_ENABLED */
00044 /* #define HAL_CAN_LEGACY_MODULE_ENABLED */
00045 /* #define HAL_DAC_MODULE_ENABLED */
00046 /* #define HAL_DCMI_MODULE_ENABLED */
00047 /* #define HAL_DMA2D_MODULE_ENABLED */
00048 /* #define HAL_ETH_MODULE_ENABLED */
00049 /* #define HAL_ETH_LEGACY_MODULE_ENABLED */
00050 /* #define HAL_NAND_MODULE_ENABLED */
00051 /* #define HAL_NOR_MODULE_ENABLED */
00052 /* #define HAL_PCCARD_MODULE_ENABLED */
00053 /* #define HAL_SRAM_MODULE_ENABLED */
00054 /* #define HAL_SDRAM_MODULE_ENABLED */
00055 /* #define HAL_HASH_MODULE_ENABLED */
00056 /* #define HAL_I2C_MODULE_ENABLED */
00057 /* #define HAL_I2S_MODULE_ENABLED */
00058 /* #define HAL_IWDG_MODULE_ENABLED */
00059 /* #define HAL_LTDC_MODULE_ENABLED */
00060 /* #define HAL_RNG_MODULE_ENABLED */
00061 /* #define HAL_RTC_MODULE_ENABLED */
00062 /* #define HAL_SAI_MODULE_ENABLED */
00063 /* #define HAL_SD_MODULE_ENABLED */
00064 /* #define HAL_MMC_MODULE_ENABLED */
00065 /* #define HAL_SPI_MODULE_ENABLED */
00066 #define HAL_TIM_MODULE_ENABLED
00067 #define HAL_UART_MODULE_ENABLED
00068 /* #define HAL_USART_MODULE_ENABLED */
00069 /* #define HAL_IRDA_MODULE_ENABLED */
00070 /* #define HAL_SMARTCARD_MODULE_ENABLED */
00071 /* #define HAL_SMBUS_MODULE_ENABLED */
00072 /* #define HAL_WWDG_MODULE_ENABLED */
00073 /* #define HAL_PCD_MODULE_ENABLED */
00074 /* #define HAL_HCD_MODULE_ENABLED */
00075 /* #define HAL_DSI_MODULE_ENABLED */
00076 /* #define HAL_QSPI_MODULE_ENABLED */
00077 /* #define HAL_QSPI_MODULE_ENABLED */
00078 /* #define HAL_CEC_MODULE_ENABLED */
00079 /* #define HAL_FMPI2C_MODULE_ENABLED */
00080 /* #define HAL_FMPMBUS_MODULE_ENABLED */
00081 /* #define HAL_SPDIFRX_MODULE_ENABLED */
00082 /* #define HAL_DFSDM_MODULE_ENABLED */
00083 /* #define HAL_LPTIM_MODULE_ENABLED */
00084 #define HAL_GPIO_MODULE_ENABLED
00085 #define HAL_EXTI_MODULE_ENABLED
00086 #define HAL_DMA_MODULE_ENABLED
00087 #define HAL_RCC_MODULE_ENABLED
00088 #define HAL_FLASH_MODULE_ENABLED
00089 #define HAL_PWR_MODULE_ENABLED
00090 #define HAL_CORTEX_MODULE_ENABLED
00091
00092 /* ##### HSE/HSI Values adaptation ##### */
00098 #if !defined (HSE_VALUE)
00099     #define HSE_VALUE      25000000U
00100 #endif /* HSE_VALUE */
00101
00102 #if !defined (HSE_STARTUP_TIMEOUT)
00103     #define HSE_STARTUP_TIMEOUT    100U
00104 #endif /* HSE_STARTUP_TIMEOUT */
00105
00111 #if !defined (HSI_VALUE)
00112     #define HSI_VALUE      ((uint32_t)16000000U)
00113 #endif /* HSI_VALUE */
00114
00118 #if !defined (LSI_VALUE)
00119     #define LSI_VALUE      32000U
00120 #endif /* LSI_VALUE */
00126 #if !defined (LSE_VALUE)
00127     #define LSE_VALUE      32768U
00128 #endif /* LSE_VALUE */
00129
00130 #if !defined (LSE_STARTUP_TIMEOUT)
00131     #define LSE_STARTUP_TIMEOUT    5000U
00132 #endif /* LSE_STARTUP_TIMEOUT */
00133
00139 #if !defined (EXTERNAL_CLOCK_VALUE)
00140     #define EXTERNAL_CLOCK_VALUE    12288000U
00141 #endif /* EXTERNAL_CLOCK_VALUE */
00142
00143 /* Tip: To avoid modifying this file each time you need to use different HSE,
00144     ==> you can define the HSE value in your toolchain compiler preprocessor. */

```

```

00145
00146 /* ##### System Configuration ##### */
00150 #define VDD_VALUE 3300U
00151 #define TICK_INT_PRIORITY 15U
00152 #define USE_RTOS 0U
00153 #define PREFETCH_ENABLE 1U
00154 #define INSTRUCTION_CACHE_ENABLE 1U
00155 #define DATA_CACHE_ENABLE 1U
00156
00157 #define USE_HAL_ADC_REGISTER_CALLBACKS 0U /* ADC register callback disabled */
00158 #define USE_HAL_CAN_REGISTER_CALLBACKS 0U /* CAN register callback disabled */
00159 #define USE_HAL_CEC_REGISTER_CALLBACKS 0U /* CEC register callback disabled */
00160 #define USE_HAL_Cryp_REGISTER_CALLBACKS 0U /* CRYPT register callback disabled */
00161 #define USE_HAL_DAC_REGISTER_CALLBACKS 0U /* DAC register callback disabled */
00162 #define USE_HAL_DCMI_REGISTER_CALLBACKS 0U /* DCMI register callback disabled */
00163 #define USE_HAL_DFSDM_REGISTER_CALLBACKS 0U /* DFSDM register callback disabled */
00164 #define USE_HAL_DMA2D_REGISTER_CALLBACKS 0U /* DMA2D register callback disabled */
00165 #define USE_HAL_DSI_REGISTER_CALLBACKS 0U /* DSI register callback disabled */
00166 #define USE_HAL_ETH_REGISTER_CALLBACKS 0U /* ETH register callback disabled */
00167 #define USE_HAL_HASH_REGISTER_CALLBACKS 0U /* HASH register callback disabled */
00168 #define USE_HAL_HCD_REGISTER_CALLBACKS 0U /* HCD register callback disabled */
00169 #define USE_HAL_I2C_REGISTER_CALLBACKS 0U /* I2C register callback disabled */
00170 #define USE_HAL_FMPI2C_REGISTER_CALLBACKS 0U /* FMPI2C register callback disabled */
00171 #define USE_HAL_FPMBSMBUS_REGISTER_CALLBACKS 0U /* FMPSMBUS register callback disabled */
00172 #define USE_HAL_I2S_REGISTER_CALLBACKS 0U /* I2S register callback disabled */
00173 #define USE_HAL_IRDA_REGISTER_CALLBACKS 0U /* IRDA register callback disabled */
00174 #define USE_HAL_LPTIM_REGISTER_CALLBACKS 0U /* LPTIM register callback disabled */
00175 #define USE_HAL_LTDC_REGISTER_CALLBACKS 0U /* LTDC register callback disabled */
00176 #define USE_HAL_MMC_REGISTER_CALLBACKS 0U /* MMC register callback disabled */
00177 #define USE_HAL_NAND_REGISTER_CALLBACKS 0U /* NAND register callback disabled */
00178 #define USE_HAL_NOR_REGISTER_CALLBACKS 0U /* NOR register callback disabled */
00179 #define USE_HAL_PCCARD_REGISTER_CALLBACKS 0U /* PCCARD register callback disabled */
00180 #define USE_HAL_PCD_REGISTER_CALLBACKS 0U /* PCD register callback disabled */
00181 #define USE_HAL_QSPI_REGISTER_CALLBACKS 0U /* QSPI register callback disabled */
00182 #define USE_HAL_RNG_REGISTER_CALLBACKS 0U /* RNG register callback disabled */
00183 #define USE_HAL_RTC_REGISTER_CALLBACKS 0U /* RTC register callback disabled */
00184 #define USE_HAL_SAI_REGISTER_CALLBACKS 0U /* SAI register callback disabled */
00185 #define USE_HAL_SD_REGISTER_CALLBACKS 0U /* SD register callback disabled */
00186 #define USE_HAL_SMARTCARD_REGISTER_CALLBACKS 0U /* SMARTCARD register callback disabled */
00187 #define USE_HAL_SDRAM_REGISTER_CALLBACKS 0U /* SDRAM register callback disabled */
00188 #define USE_HAL_SRAM_REGISTER_CALLBACKS 0U /* SRAM register callback disabled */
00189 #define USE_HAL_SPDIFRX_REGISTER_CALLBACKS 0U /* SPDIFRX register callback disabled */
00190 #define USE_HAL_SMBUS_REGISTER_CALLBACKS 0U /* SMBUS register callback disabled */
00191 #define USE_HAL_SPI_REGISTER_CALLBACKS 0U /* SPI register callback disabled */
00192 #define USE_HAL_TIM_REGISTER_CALLBACKS 0U /* TIM register callback disabled */
00193 #define USE_HAL_UART_REGISTER_CALLBACKS 0U /* UART register callback disabled */
00194 #define USE_HAL_USART_REGISTER_CALLBACKS 0U /* USART register callback disabled */
00195 #define USE_HAL_WWDG_REGISTER_CALLBACKS 0U /* WWDG register callback disabled */
00196
00197 /* ##### Assert Selection ##### */
00202 #define USE_FULL_ASSERT 1U /*
00203
00204 /* ##### Ethernet peripheral configuration ##### */
00205
00206 /* Section 1 : Ethernet peripheral configuration */
00207
00208 /* MAC ADDRESS: MAC_ADDR0:MAC_ADDR1:MAC_ADDR2:MAC_ADDR3:MAC_ADDR4:MAC_ADDR5 */
00209 #define MAC_ADDR0 2U
00210 #define MAC_ADDR1 0U
00211 #define MAC_ADDR2 0U
00212 #define MAC_ADDR3 0U
00213 #define MAC_ADDR4 0U
00214 #define MAC_ADDR5 0U
00215
00216 /* Definition of the Ethernet driver buffers size and count */
00217 #define ETH_RX_BUF_SIZE /* buffer size for receive */
00218 #define ETH_TX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for transmit */
00219 #define ETH_RXBUFNB 4U /* 4 Rx buffers of size ETH_RX_BUF_SIZE */
00220 #define ETH_TXBUFNB 4U /* 4 Tx buffers of size ETH_TX_BUF_SIZE */
00221
00222 /* Section 2: PHY configuration section */
00223
00224 /* DP83848_PHY_ADDRESS Address*/
00225 #define DP83848_PHY_ADDRESS 0x01U
00226 /* PHY Reset delay these values are based on a 1 ms SysTick interrupt*/
00227 #define PHY_RESET_DELAY 0x000000FFU
00228 /* PHY Configuration delay */
00229 #define PHY_CONFIG_DELAY 0x000000FFU
00230
00231 #define PHY_READ_TO 0x000000FFU
00232 #define PHY_WRITE_TO 0x000000FFU
00233
00234 /* Section 3: Common PHY Registers */
00235
00236 #define PHY_BCR ((uint16_t)0x0000U)
00237 #define PHY_BSR ((uint16_t)0x0001U)
00239 #define PHY_RESET ((uint16_t)0x8000U)

```

```

00240 #define PHY_LOOPBACK                ((uint16_t)0x4000U)
00241 #define PHY_FULDDUPLEX_100M          ((uint16_t)0x2100U)
00242 #define PHY_HALFDUPLEX_100M          ((uint16_t)0x2000U)
00243 #define PHY_FULDDUPLEX_10M           ((uint16_t)0x0100U)
00244 #define PHY_HALFDUPLEX_10M           ((uint16_t)0x0000U)
00245 #define PHY_AUTONEGOTIATION           ((uint16_t)0x1000U)
00246 #define PHY_RESTART_AUTONEGOTIATION  ((uint16_t)0x0200U)
00247 #define PHY_POWERDOWN                 ((uint16_t)0x0800U)
00248 #define PHY_ISOLATE                   ((uint16_t)0x0400U)
00250 #define PHY_AUTONEGO_COMPLETE         ((uint16_t)0x0020U)
00251 #define PHY_LINKED_STATUS              ((uint16_t)0x0004U)
00252 #define PHY_JABBER_DETECTION           ((uint16_t)0x0002U)
00254 /* Section 4: Extended PHY Registers */
00255 #define PHY_SR                         ((uint16_t)0x10U)
00257 #define PHY_SPEED_STATUS               ((uint16_t)0x0002U)
00258 #define PHY_DUPLEX_STATUS              ((uint16_t)0x0004U)
00260 /* ##### SPI peripheral configuration ##### */
00261
00262 /* CRC FEATURE: Use to activate CRC feature inside HAL SPI Driver
00263 * Activated: CRC code is present inside driver
00264 * Deactivated: CRC code cleaned from driver
00265 */
00266
00267 #define USE_SPI_CRC                     0U
00268
00269 /* Includes -----*/
00274 #ifndef HAL_RCC_MODULE_ENABLED
00275     #include "stm32f4xx_hal_rcc.h"
00276 #endif /* HAL_RCC_MODULE_ENABLED */
00277
00278 #ifndef HAL_GPIO_MODULE_ENABLED
00279     #include "stm32f4xx_hal_gpio.h"
00280 #endif /* HAL_GPIO_MODULE_ENABLED */
00281
00282 #ifndef HAL_EXTI_MODULE_ENABLED
00283     #include "stm32f4xx_hal_exti.h"
00284 #endif /* HAL_EXTI_MODULE_ENABLED */
00285
00286 #ifndef HAL_DMA_MODULE_ENABLED
00287     #include "stm32f4xx_hal_dma.h"
00288 #endif /* HAL_DMA_MODULE_ENABLED */
00289
00290 #ifndef HAL_CORTEX_MODULE_ENABLED
00291     #include "stm32f4xx_hal_cortex.h"
00292 #endif /* HAL_CORTEX_MODULE_ENABLED */
00293
00294 #ifndef HAL_ADC_MODULE_ENABLED
00295     #include "stm32f4xx_hal_adc.h"
00296 #endif /* HAL_ADC_MODULE_ENABLED */
00297
00298 #ifndef HAL_CAN_MODULE_ENABLED
00299     #include "stm32f4xx_hal_can.h"
00300 #endif /* HAL_CAN_MODULE_ENABLED */
00301
00302 #ifndef HAL_CAN_LEGACY_MODULE_ENABLED
00303     #include "stm32f4xx_hal_can_legacy.h"
00304 #endif /* HAL_CAN_LEGACY_MODULE_ENABLED */
00305
00306 #ifndef HAL_CRC_MODULE_ENABLED
00307     #include "stm32f4xx_hal_crc.h"
00308 #endif /* HAL_CRC_MODULE_ENABLED */
00309
00310 #ifndef HAL_CRYP_MODULE_ENABLED
00311     #include "stm32f4xx_hal_cryp.h"
00312 #endif /* HAL_CRYP_MODULE_ENABLED */
00313
00314 #ifndef HAL_DMA2D_MODULE_ENABLED
00315     #include "stm32f4xx_hal_dma2d.h"
00316 #endif /* HAL_DMA2D_MODULE_ENABLED */
00317
00318 #ifndef HAL_DAC_MODULE_ENABLED
00319     #include "stm32f4xx_hal_dac.h"
00320 #endif /* HAL_DAC_MODULE_ENABLED */
00321
00322 #ifndef HAL_DCMI_MODULE_ENABLED
00323     #include "stm32f4xx_hal_dcmi.h"
00324 #endif /* HAL_DCMI_MODULE_ENABLED */
00325
00326 #ifndef HAL_ETH_MODULE_ENABLED
00327     #include "stm32f4xx_hal_eth.h"
00328 #endif /* HAL_ETH_MODULE_ENABLED */
00329
00330 #ifndef HAL_ETH_LEGACY_MODULE_ENABLED
00331     #include "stm32f4xx_hal_eth_legacy.h"
00332 #endif /* HAL_ETH_LEGACY_MODULE_ENABLED */
00333
00334 #ifndef HAL_FLASH_MODULE_ENABLED

```

```
00335 #include "stm32f4xx_hal_flash.h"
00336 #endif /* HAL_FLASH_MODULE_ENABLED */
00337
00338 #ifdef HAL_SRAM_MODULE_ENABLED
00339 #include "stm32f4xx_hal_sram.h"
00340 #endif /* HAL_SRAM_MODULE_ENABLED */
00341
00342 #ifdef HAL_NOR_MODULE_ENABLED
00343 #include "stm32f4xx_hal_nor.h"
00344 #endif /* HAL_NOR_MODULE_ENABLED */
00345
00346 #ifdef HAL_NAND_MODULE_ENABLED
00347 #include "stm32f4xx_hal_nand.h"
00348 #endif /* HAL_NAND_MODULE_ENABLED */
00349
00350 #ifdef HAL_PCCARD_MODULE_ENABLED
00351 #include "stm32f4xx_hal_pccard.h"
00352 #endif /* HAL_PCCARD_MODULE_ENABLED */
00353
00354 #ifdef HAL_SDRAM_MODULE_ENABLED
00355 #include "stm32f4xx_hal_sdram.h"
00356 #endif /* HAL_SDRAM_MODULE_ENABLED */
00357
00358 #ifdef HAL_HASH_MODULE_ENABLED
00359 #include "stm32f4xx_hal_hash.h"
00360 #endif /* HAL_HASH_MODULE_ENABLED */
00361
00362 #ifdef HAL_I2C_MODULE_ENABLED
00363 #include "stm32f4xx_hal_i2c.h"
00364 #endif /* HAL_I2C_MODULE_ENABLED */
00365
00366 #ifdef HAL_SMBUS_MODULE_ENABLED
00367 #include "stm32f4xx_hal_smbus.h"
00368 #endif /* HAL_SMBUS_MODULE_ENABLED */
00369
00370 #ifdef HAL_I2S_MODULE_ENABLED
00371 #include "stm32f4xx_hal_i2s.h"
00372 #endif /* HAL_I2S_MODULE_ENABLED */
00373
00374 #ifdef HAL_IWDG_MODULE_ENABLED
00375 #include "stm32f4xx_hal_iwdg.h"
00376 #endif /* HAL_IWDG_MODULE_ENABLED */
00377
00378 #ifdef HAL_LTDC_MODULE_ENABLED
00379 #include "stm32f4xx_hal_ltdc.h"
00380 #endif /* HAL_LTDC_MODULE_ENABLED */
00381
00382 #ifdef HAL_PWR_MODULE_ENABLED
00383 #include "stm32f4xx_hal_pwr.h"
00384 #endif /* HAL_PWR_MODULE_ENABLED */
00385
00386 #ifdef HAL_RNG_MODULE_ENABLED
00387 #include "stm32f4xx_hal_rng.h"
00388 #endif /* HAL_RNG_MODULE_ENABLED */
00389
00390 #ifdef HAL_RTC_MODULE_ENABLED
00391 #include "stm32f4xx_hal_rtc.h"
00392 #endif /* HAL_RTC_MODULE_ENABLED */
00393
00394 #ifdef HAL_SAI_MODULE_ENABLED
00395 #include "stm32f4xx_hal_sai.h"
00396 #endif /* HAL_SAI_MODULE_ENABLED */
00397
00398 #ifdef HAL_SD_MODULE_ENABLED
00399 #include "stm32f4xx_hal_sd.h"
00400 #endif /* HAL_SD_MODULE_ENABLED */
00401
00402 #ifdef HAL_SPI_MODULE_ENABLED
00403 #include "stm32f4xx_hal_spi.h"
00404 #endif /* HAL_SPI_MODULE_ENABLED */
00405
00406 #ifdef HAL_TIM_MODULE_ENABLED
00407 #include "stm32f4xx_hal_tim.h"
00408 #endif /* HAL_TIM_MODULE_ENABLED */
00409
00410 #ifdef HAL_UART_MODULE_ENABLED
00411 #include "stm32f4xx_hal_uart.h"
00412 #endif /* HAL_UART_MODULE_ENABLED */
00413
00414 #ifdef HAL_USART_MODULE_ENABLED
00415 #include "stm32f4xx_hal_usart.h"
00416 #endif /* HAL_USART_MODULE_ENABLED */
00417
00418 #ifdef HAL_IRDA_MODULE_ENABLED
00419 #include "stm32f4xx_hal_irda.h"
00420 #endif /* HAL_IRDA_MODULE_ENABLED */
00421
```

```

00422 #ifdef HAL_SMARTCARD_MODULE_ENABLED
00423 #include "stm32f4xx_hal_smartcard.h"
00424 #endif /* HAL_SMARTCARD_MODULE_ENABLED */
00425
00426 #ifdef HAL_WWDG_MODULE_ENABLED
00427 #include "stm32f4xx_hal_wwdg.h"
00428 #endif /* HAL_WWDG_MODULE_ENABLED */
00429
00430 #ifdef HAL_PCD_MODULE_ENABLED
00431 #include "stm32f4xx_hal_pcd.h"
00432 #endif /* HAL_PCD_MODULE_ENABLED */
00433
00434 #ifdef HAL_HCD_MODULE_ENABLED
00435 #include "stm32f4xx_hal_hcd.h"
00436 #endif /* HAL_HCD_MODULE_ENABLED */
00437
00438 #ifdef HAL_DSI_MODULE_ENABLED
00439 #include "stm32f4xx_hal_dsi.h"
00440 #endif /* HAL_DSI_MODULE_ENABLED */
00441
00442 #ifdef HAL_QSPI_MODULE_ENABLED
00443 #include "stm32f4xx_hal_qspi.h"
00444 #endif /* HAL_QSPI_MODULE_ENABLED */
00445
00446 #ifdef HAL_CEC_MODULE_ENABLED
00447 #include "stm32f4xx_hal_cec.h"
00448 #endif /* HAL_CEC_MODULE_ENABLED */
00449
00450 #ifdef HAL_FMPI2C_MODULE_ENABLED
00451 #include "stm32f4xx_hal_fmpi2c.h"
00452 #endif /* HAL_FMPI2C_MODULE_ENABLED */
00453
00454 #ifdef HAL_FMPMBUS_MODULE_ENABLED
00455 #include "stm32f4xx_hal_fmpmbus.h"
00456 #endif /* HAL_FMPMBUS_MODULE_ENABLED */
00457
00458 #ifdef HAL_SPDIFRX_MODULE_ENABLED
00459 #include "stm32f4xx_hal_spdifrx.h"
00460 #endif /* HAL_SPDIFRX_MODULE_ENABLED */
00461
00462 #ifdef HAL_DFSDM_MODULE_ENABLED
00463 #include "stm32f4xx_hal_dfsdm.h"
00464 #endif /* HAL_DFSDM_MODULE_ENABLED */
00465
00466 #ifdef HAL_LPTIM_MODULE_ENABLED
00467 #include "stm32f4xx_hal_lptim.h"
00468 #endif /* HAL_LPTIM_MODULE_ENABLED */
00469
00470 #ifdef HAL_MMC_MODULE_ENABLED
00471 #include "stm32f4xx_hal_mmc.h"
00472 #endif /* HAL_MMC_MODULE_ENABLED */
00473
00474 /* Exported macro -----*/
00475 #ifdef USE_FULL_ASSERT
00476 #define assert_param(expr) ((expr) ? (void)0U : assert_failed((uint8_t *)__FILE__, __LINE__))
00477 /* Exported functions ----- */
00478 void assert_failed(uint8_t* file, uint32_t line);
00479 #else
00480 #define assert_param(expr) ((void)0U)
00481 #endif /* USE_FULL_ASSERT */
00482
00483 #ifdef __cplusplus
00484 }
00485 #endif
00486
00487 #endif /* __STM32F4xx_HAL_CONF_H */

```

6.16 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Inc/stm32f4xx_it.h File Reference

This file contains the headers of the interrupt handlers.

Functions

- void **NMI_Handler** (void)

This function handles Non maskable interrupt.

- void **HardFault_Handler** (void)

This function handles Hard fault interrupt.

- void **MemManage_Handler** (void)

This function handles Memory management fault.

- void **BusFault_Handler** (void)

This function handles Pre-fetch fault, memory access fault.

- void **UsageFault_Handler** (void)

This function handles Undefined instruction or illegal state.

- void **DebugMon_Handler** (void)

This function handles Debug monitor.

- void **TIM1_BRK_TIM9_IRQHandler** (void)

This function handles TIM1 break interrupt and TIM9 global interrupt.

6.16.1 Detailed Description

This file contains the headers of the interrupt handlers.

Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

6.17 stm32f4xx_it.h

[Go to the documentation of this file.](#)

```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020 /* Define to prevent recursive inclusion -----*/
00021 #ifndef __STM32F4xx_IT_H
00022 #define __STM32F4xx_IT_H
00023
00024 #ifdef __cplusplus
00025     extern "C" {
00026 #endif
00027
00028 /* Private includes -----*/
00029 /* USER CODE BEGIN Includes */
00030
00031 /* USER CODE END Includes */
00032
00033 /* Exported types -----*/
00034 /* USER CODE BEGIN ET */
00035
00036 /* USER CODE END ET */
00037
00038 /* Exported constants -----*/
00039 /* USER CODE BEGIN EC */
00040
00041 /* USER CODE END EC */
00042
00043 /* Exported macro -----*/
00044 /* USER CODE BEGIN EM */
00045
00046 /* USER CODE END EM */
00047
00048 /* Exported functions prototypes -----*/
```

```

00049 void NMI_Handler(void);
00050 void HardFault_Handler(void);
00051 void MemManage_Handler(void);
00052 void BusFault_Handler(void);
00053 void UsageFault_Handler(void);
00054 void DebugMon_Handler(void);
00055 void TIM1_BRK_TIM9_IRQHandler(void);
00056 /* USER CODE BEGIN EFP */
00057
00058 /* USER CODE END EFP */
00059
00060 #ifdef __cplusplus
00061 }
00062 #endif
00063
00064 #endif /* __STM32F4xx_IT_H */

```

6.18 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/console.c File Reference

Command Line Interpreter based on FreeRTOS and STM32 HAL layer Github account: <https://github.com/aaron-ev>.

```

#include "FreeRTOS.h"
#include "FreeRTOS_CLI.h"
#include "task.h"
#include "queue.h"
#include "stm32f401xc.h"
#include "stm32f4xx_hal.h"
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include "bsp.h"

```

Macros

- #define **MAX_IN_STR_LEN** 300
- #define **MAX_OUT_STR_LEN** 600
- #define **MAX_RX_QUEUE_LEN** 300
- #define **ASCII_TAB** '\t' /* Tabulate */
- #define **ASCII_CR** '\r' /* Carriage return */
- #define **ASCII_LF** '\n' /* Line feed */
- #define **ASCII_BACKSPACE** '\b' /* Back space */
- #define **ASCII_FORM_FEED** '\f' /* Form feed */
- #define **ASCII_DEL** 127 /* Delete */
- #define **ASCII_CTRL_PLUS_C** 3 /* CTRL + C */
- #define **ASCII_NACK** 21 /* Negative acknowledge */

Functions

- static BaseType_t **prvCommandPwmSetFreq** (char *pcWriteBuffer, size_t xWriteBufferLen, const char *pcCommandString)
Command that sets a new pwm frequency.
- static BaseType_t **prvCommandPwmSetDuty** (char *pcWriteBuffer, size_t xWriteBufferLen, const char *pcCommandString)
Command that sets a new pwm duty cycle.

- static BaseType_t [prvCommandGpioWrite](#) (char *pcWriteBuffer, size_t xWriteBufferLen, const char *pcCommandString)
Command that writes to a GPIO pin.
- static BaseType_t [prvCommandGpioRead](#) (char *pcWriteBuffer, size_t xWriteBufferLen, const char *pcCommandString)
Command that reads from GPIO Pin.
- static BaseType_t [prvCommandEcho](#) (char *pcWriteBuffer, size_t xWriteBufferLen, const char *pcCommandString)
Echo command line in UNIX systems.
- static BaseType_t [prvCommandTaskStats](#) (char *pcWriteBuffer, size_t xWriteBufferLen, const char *pcCommandString)
Command that gets task statistics.
- static BaseType_t [prvCommandHeap](#) (char *pcWriteBuffer, size_t xWriteBufferLen, const char *pcCommandString)
Command that gets heap information.
- static BaseType_t [prvCommandClik](#) (char *pcWriteBuffer, size_t xWriteBufferLen, const char *pcCommandString)
Command that gets clock information from current clock settings.
- static BaseType_t [prvCommandTicks](#) (char *pcWriteBuffer, size_t xWriteBufferLen, const char *pcCommandString)
Command that calculate OS ticks information.
- static const char * [prvpcMapTaskState](#) (eTaskState eState)
This function is executed in case of error occurrence.
- static BaseType_t [xConsoleRead](#) (uint8_t *cReadChar)
Reads from UART RX buffer. Reads one byte at the time.
- static HAL_StatusTypeDef [vConsoleWrite](#) (const char *buff)
Write to UART TX.
- void [vConsoleEnableRxInterrupt](#) (void)
Enables UART RX reception.
- void [vTaskConsole](#) (void *pvParams)
Task to handle user commands via serial communication.
- BaseType_t [xConsoleInit](#) (uint16_t usStackSize, UBaseType_t uxPriority, UART_HandleTypeDef *pxUartHandle)
Initialize the console by registering all commands and creating a task.
- void [HAL_UART_RxCpltCallback](#) (UART_HandleTypeDef *huart)
Callback for UART RX, triggered any time there is a new character.

Variables

- char **cRxData**
- QueueHandle_t **xQueueRxHandle**
- UART_HandleTypeDef * **pxUartDevHandle**
- static const char * **pcWelcomeMsg** = "Welcome to the console. Enter 'help' to view a list of available commands.\n"
- static const char * [prvpcTaskListHeader](#)
- static const char * **prvpcPrompt** = "#cmd: "
- static const CLI_Command_Definition_t **xCommands** []

6.18.1 Detailed Description

Command Line Interpreter based on FreeRTOS and STM32 HAL layer Github account: <https://github.com/aaron-ev>.

Author

Aaron Escoboza

6.18.2 Function Documentation

6.18.2.1 HAL_UART_RxCpltCallback()

```
void HAL_UART_RxCpltCallback (
    UART_HandleTypeDef * huart )
```

Callback for UART RX, triggered any time there is a new character.

Parameters

<i>*huart</i>	Pointer to the uart handle.
---------------	-----------------------------

Return values

<i>void</i>	
-------------	--

6.18.2.2 prvCommandClk()

```
static BaseType_t prvCommandClk (
    char * pcWriteBuffer,
    size_t xWriteBufferLen,
    const char * pcCommandString ) [static]
```

Command that gets clock information from current clock settings.

Parameters

<i>*pcWriteBuffer</i>	FreeRTOS CLI write buffer.
<i>xWriteBufferLen</i>	Length of write buffer.
<i>*pcCommandString</i>	pointer to the command name.

Return values

<i>HAL</i>	status
------------	--------

6.18.2.3 prvCommandEcho()

```
static BaseType_t prvCommandEcho (
    char * pcWriteBuffer,
    size_t xWriteBufferLen,
    const char * pcCommandString ) [static]
```

Echo command line in UNIX systems.

Parameters

<i>*pcWriteBuffer</i>	FreeRTOS CLI write buffer.
<i>xWriteBufferLen</i>	Length of write buffer.
<i>*pcCommandString</i>	pointer to the command name.

Return values

<i>HAL</i>	status
------------	--------

6.18.2.4 prvCommandGpioRead()

```
static BaseType_t prvCommandGpioRead (
    char * pcWriteBuffer,
    size_t xWriteBufferLen,
    const char * pcCommandString ) [static]
```

Command that reads from GPIOx Pin.

Parameters

<i>*pcWriteBuffer</i>	FreeRTOS CLI write buffer.
<i>xWriteBufferLen</i>	Length of write buffer.
<i>*pcCommandString</i>	pointer to the command name.

Return values

<i>HAL</i>	status
------------	--------

6.18.2.5 prvCommandGpioWrite()

```
static BaseType_t prvCommandGpioWrite (
    char * pcWriteBuffer,
    size_t xWriteBufferLen,
    const char * pcCommandString ) [static]
```

Command that writes to a GPIOx pin.

Parameters

<i>*pcWriteBuffer</i>	FreeRTOS CLI write buffer.
<i>xWriteBufferLen</i>	Length of write buffer.
<i>*pcCommandString</i>	pointer to the command name.

Return values

<i>HAL</i>	status
------------	--------

6.18.2.6 prvCommandHeap()

```
static BaseType_t prvCommandHeap (
    char * pcWriteBuffer,
    size_t xWriteBufferLen,
    const char * pcCommandString ) [static]
```

Command that gets heap information.

Parameters

<i>*pcWriteBuffer</i>	FreeRTOS CLI write buffer.
<i>xWriteBufferLen</i>	Length of write buffer.
<i>*pcCommandString</i>	pointer to the command name.

Return values

<i>HAL</i>	status
------------	--------

6.18.2.7 prvCommandPwmSetDuty()

```
static BaseType_t prvCommandPwmSetDuty (
    char * pcWriteBuffer,
    size_t xWriteBufferLen,
    const char * pcCommandString ) [static]
```

Command that sets a new pwm duty cycle.

Parameters

<i>*pcWriteBuffer</i>	FreeRTOS CLI write buffer.
<i>xWriteBufferLen</i>	Length of write buffer.
<i>*pcCommandString</i>	pointer to the command name.

Return values

<i>HAL</i>	status
------------	--------

6.18.2.8 prvCommandPwmSetFreq()

```
static BaseType_t prvCommandPwmSetFreq (
    char * pcWriteBuffer,
    size_t xWriteBufferLen,
    const char * pcCommandString ) [static]
```

Command that sets a new pwm frequency.

Parameters

<i>*pcWriteBuffer</i>	FreeRTOS CLI write buffer.
<i>xWriteBufferLen</i>	Length of write buffer.
<i>*pcCommandString</i>	pointer to the command name.

Return values

<i>HAL</i>	status
------------	--------

6.18.2.9 prvCommandTaskStats()

```
static BaseType_t prvCommandTaskStats (
    char * pcWriteBuffer,
    size_t xWriteBufferLen,
    const char * pcCommandString ) [static]
```

Command that gets task statistics.

Parameters

<i>*pcWriteBuffer</i>	FreeRTOS CLI write buffer.
<i>xWriteBufferLen</i>	Length of write buffer.
<i>*pcCommandString</i>	pointer to the command name.

Return values

<i>HAL</i>	status
------------	--------

6.18.2.10 prvCommandTicks()

```
static BaseType_t prvCommandTicks (
    char * pcWriteBuffer,
    size_t xWriteBufferLen,
    const char * pcCommandString ) [static]
```

Command that calculate OS ticks information.

Parameters

<i>*pcWriteBuffer</i>	FreeRTOS CLI write buffer.
<i>xWriteBufferLen</i>	Length of write buffer.
<i>*pcCommandString</i>	pointer to the command name.

Return values

<i>HAL</i>	status
------------	--------

6.18.2.11 prvpcMapTaskState()

```
static const char * prvpcMapTaskState (
    eTaskState eState ) [static]
```

This function is executed in case of error occurrence.

Return values

<i>None</i>	
-------------	--

6.18.2.12 vConsoleEnableRxInterrupt()

```
void vConsoleEnableRxInterrupt (
    void )
```

Enables UART RX reception.

Parameters

<i>void</i>	
-------------	--

Return values

<i>void</i>	
-------------	--

6.18.2.13 vConsoleWrite()

```
static HAL_StatusTypeDef vConsoleWrite (
    const char * buff ) [static]
```

Write to UART TX.

Parameters

<i>*buff</i>	buffer to be written.
--------------	-----------------------

Return values

<i>HAL</i>	status
------------	--------

6.18.2.14 vTaskConsole()

```
void vTaskConsole (
    void * pvParams )
```

Task to handle user commands via serial communication.

Parameters

<i>void*</i>	Data passed at task creation.
--------------	-------------------------------

Return values

<i>void</i>	
-------------	--

6.18.2.15 xConsoleInit()

```
BaseType_t xConsoleInit (
    uint16_t usStackSize,
    UBaseType_t uxPriority,
    UART_HandleTypeDef * pxUartHandle )
```

Initialize the console by registering all commands and creating a task.

Parameters

<i>usStackSize</i>	Task console stack size
<i>uxPriority</i>	Task console priority
<i>*pxUartHandle</i>	Pointer for uart handle.

Return values

<i>FreeRTOS</i>	status
-----------------	--------

6.18.2.16 xConsoleRead()

```
static BaseType_t xConsoleRead (
    uint8_t * cReadChar ) [static]
```

Reads from UART RX buffer. Reads one byte at the time.

Parameters

<code>*cReadChar</code>	pointer to where data will be stored.
-------------------------	---------------------------------------

Return values

<code>HAL</code>	status
------------------	--------

6.18.3 Variable Documentation

6.18.3.1 prvpcTaskListHeader

```
const char* prvpcTaskListHeader [static]
```

Initial value:

```
= "Task states: BL = Blocked RE = Ready DE = Deleted SU = Suspended\n\n"
    "Task name                               State Priority Stack remaining\n"
    "%CPU usage Runtime\n"
    "===== \n"
```

6.19 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFree↵ RTOS/Core/Src/main.c File Reference

Command Line Interpreter based on FreeRTOS and STM32 HAL layer Github account: <https://github.com/aaron-ev>.

```
#include "stm32f401xc.h"
#include "stm32f4xx_hal.h"
#include "appConfig.h"
#include "FreeRTOS.h"
#include "task.h"
#include "bsp.h"
#include "console.h"
```

Functions

- void [Error_Handler](#) (void)
This function is executed in case of error occurrence.
- void **vTaskHeartBeat** (void *params)
- int **main** (void)
- void [HAL_TIM_PeriodElapsedCallback](#) (TIM_HandleTypeDef *htim)
Period elapsed callback in non blocking mode.

Variables

- TaskHandle_t **xTaskHeartBeatHandler**
- UART_HandleTypeDef **consoleHandle**

6.19.1 Detailed Description

Command Line Interpreter based on FreeRTOS and STM32 HAL layer Github account: <https://github.com/aaron-ev>.

Author

Aaron Escoboza

6.19.2 Function Documentation

6.19.2.1 Error_Handler()

```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

Return values

None	
------	--

6.19.2.2 HAL_TIM_PeriodElapsedCallback()

```
void HAL_TIM_PeriodElapsedCallback (
    TIM_HandleTypeDef * htim )
```

Period elapsed callback in non blocking mode.

Note

This function is called when TIM9 interrupt took place, inside HAL_TIM_IRQHandler(). It makes a direct call to HAL_IncTick() to increment a global variable "uwTick" used as application time base.

Parameters

<i>htim</i>	: TIM handle
-------------	--------------

Return values

None	
------	--

6.20 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/stm32f4xx_hal_timebase_tim.c File Reference

HAL time base based on the hardware TIM.

```
#include "stm32f4xx_hal.h"
#include "stm32f4xx_hal_tim.h"
```

Functions

- HAL_StatusTypeDef [HAL_InitTick](#) (uint32_t TickPriority)
This function configures the TIM9 as a time base source. The time source is configured to have 1ms time base with a dedicated Tick interrupt priority.
- void [HAL_SuspendTick](#) (void)
Suspend Tick increment.
- void [HAL_ResumeTick](#) (void)
Resume Tick increment.

Variables

- TIM_HandleTypeDef **htim9**

6.20.1 Detailed Description

HAL time base based on the hardware TIM.

Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

6.20.2 Function Documentation

6.20.2.1 HAL_InitTick()

```
HAL_StatusTypeDef HAL_InitTick (
    uint32_t TickPriority )
```

This function configures the TIM9 as a time base source. The time source is configured to have 1ms time base with a dedicated Tick interrupt priority.

Note

This function is called automatically at the beginning of program after reset by HAL_Init() or at any time when clock is configured, by HAL_RCC_ClockConfig().

Parameters

<i>TickPriority</i>	Tick interrupt priority.
---------------------	--------------------------

Return values

<i>HAL</i>	status
------------	--------

6.20.2.2 HAL_ResumeTick()

```
void HAL_ResumeTick (
    void )
```

Resume Tick increment.

Note

Enable the tick increment by Enabling TIM9 update interrupt.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

6.20.2.3 HAL_SuspendTick()

```
void HAL_SuspendTick (
    void )
```

Suspend Tick increment.

Note

Disable the tick increment by disabling TIM9 update interrupt.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

6.21 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/syscalls.c File Reference

STM32CubeIDE Minimal System calls file.

```
#include <sys/stat.h>
#include <stdlib.h>
#include <errno.h>
#include <stdio.h>
#include <signal.h>
#include <time.h>
#include <sys/time.h>
#include <sys/times.h>
```

Functions

- int **__io_putchar** (int ch) **__attribute__((weak))**
- int **__io_getchar** (void)
- void **initialise_monitor_handles** ()
- int **_getpid** (void)
- int **_kill** (int pid, int sig)
- void **_exit** (int status)
- **__attribute__((weak))**
- int **_close** (int file)
- int **_fstat** (int file, struct stat *st)
- int **_isatty** (int file)
- int **_lseek** (int file, int ptr, int dir)
- int **_open** (char *path, int flags,...)
- int **_wait** (int *status)
- int **_unlink** (char *name)
- int **_times** (struct tms *buf)
- int **_stat** (char *file, struct stat *st)
- int **_link** (char *old, char *new)
- int **_fork** (void)
- int **_execve** (char *name, char **argv, char **env)

Variables

- char ** **environ** = __env

6.21.1 Detailed Description

STM32CubeIDE Minimal System calls file.

Author

Auto-generated by STM32CubeIDE

For more information about which c-functions
need which of these lowlevel functions
please consult the Newlib libc-manual

Attention

Copyright (c) 2020-2023 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

6.22 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/systemem.c File Reference

STM32CubeIDE System Memory calls file.

```
#include <errno.h>
#include <stdint.h>
```

Functions

- void * [_sbrk](#) (ptrdiff_t incr)
[_sbrk\(\)](#) allocates memory to the newlib heap and is used by malloc and others from the C library

Variables

- static uint8_t * [__sbrk_heap_end](#) = NULL

6.22.1 Detailed Description

STM32CubeIDE System Memory calls file.

Author

Generated by STM32CubeIDE

For more information about which C functions
need which of these lowlevel functions
please consult the newlib libc manual

Attention

Copyright (c) 2023 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

6.22.2 Function Documentation

6.22.2.1 [_sbrk\(\)](#)

```
void * _sbrk (
    ptrdiff_t incr )
```

[_sbrk\(\)](#) allocates memory to the newlib heap and is used by malloc and others from the C library

```
* #####
* # .data # .bss #          newlib heap          #          MSP stack          #
* #          #          #          #          # Reserved by _Min_Stack_Size #
* #####
* ^-- RAM start          ^-- _end          _estack, RAM end --^
*
```

This implementation starts allocating at the '_end' linker symbol The '_Min_Stack_Size' linker symbol reserves a memory for the MSP stack The implementation considers '_estack' linker symbol to be RAM end NOTE: If the MSP stack, at any point during execution, grows larger than the reserved size, please increase the '_Min_Stack_Size'.

Parameters

<i>incr</i>	Memory size
-------------	-------------

Returns

Pointer to allocated memory

6.22.3 Variable Documentation

6.22.3.1 __sbrk_heap_end

```
uint8_t* __sbrk_heap_end = NULL [static]
```

Pointer to the current high watermark of the heap usage

6.23 C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/Src/system_stm32f4xx.c File Reference

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File.

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

Macros

- `#define HSE_VALUE ((uint32_t)25000000)`
- `#define HSI_VALUE ((uint32_t)16000000)`

Functions

- void `SystemInit` (void)
Setup the microcontroller system Initialize the FPU setting, vector table location and External memory configuration.
- 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 = 16000000`
- `const uint8_t AHBPrescTable [16] = {0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9}`
- `const uint8_t APBPrescTable [8] = {0, 0, 0, 0, 1, 2, 3, 4}`

6.23.1 Detailed Description

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File.

Author

MCD Application Team

This file provides two functions and one global variable to be called from user application:

- [SystemInit\(\)](#): 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.

Attention

Copyright (c) 2017 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

Index

`__sbrk_heap_end`
 `sysmem.c`, [48](#)
`_sbrk`
 `sysmem.c`, [47](#)

`bsp.c`
 `bspInit`, [18](#)
 `clkInit`, [18](#)
 `consoleInit`, [19](#)
 `heartBeatInit`, [19](#)
`bsp.h`
 `bspInit`, [15](#)
`bspClk.c`
 `bspGetClockInfo`, [20](#)
`bspGetClockInfo`
 `bspClk.c`, [20](#)
`bspGpio.c`
 `bspGpioMapInstance`, [21](#)
 `bspGpioRead`, [21](#)
 `bspGpioToggle`, [21](#)
 `bspGpioWrite`, [22](#)
 `bspMapInstanceToHal`, [22](#)
 `bspMapPinNumFromBspToHal`, [22](#)
`bspGpioMapInstance`
 `bspGpio.c`, [21](#)
`bspGpioRead`
 `bspGpio.c`, [21](#)
`bspGpioToggle`
 `bspGpio.c`, [21](#)
`bspGpioWrite`
 `bspGpio.c`, [22](#)
`bspInit`
 `bsp.c`, [18](#)
 `bsp.h`, [15](#)
`bspMapInstanceToHal`
 `bspGpio.c`, [22](#)
`bspMapPinNumFromBspToHal`
 `bspGpio.c`, [22](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[15](#), [16](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[16](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[16](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[17](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core/bsp/src/bsp.c`
[17](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[19](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[20](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[23](#), [24](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[24](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[24](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[26](#), [27](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[27](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[32](#), [33](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[34](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[42](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[43](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[46](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[47](#)

`C:/Users/aaron/main/projects/cli_freeRTOS/workspace/cliFreeRTOS/Core`
[48](#)

`clkInit`
 `bsp.c`, [18](#)

`CMSIS`, [7](#)

`console.c`
 `HAL_UART_RxCpltCallback`, [36](#)
 `prvCommandClk`, [36](#)
 `prvCommandEcho`, [36](#)
 `prvCommandGpioRead`, [37](#)
 `prvCommandGpioWrite`, [37](#)
 `prvCommandHeap`, [38](#)
 `prvCommandPwmSetDuty`, [38](#)
 `prvCommandPwmSetFreq`, [39](#)
 `prvCommandTaskStats`, [39](#)
 `prvCommandTicks`, [39](#)
 `prvpcMapTaskState`, [40](#)
 `prvpcTaskListHeader`, [42](#)
 `vConsoleEnableRxInterrupt`, [40](#)
 `vConsoleWrite`, [40](#)
 `vTaskConsole`, [41](#)
 `xConsoleInit`, [41](#)
 `xConsoleRead`, [41](#)

consoleInit
 bsp.c, 19

Error_Handler
 main.c, 43
 main.h, 26

HAL_InitTick
 stm32f4xx_hal_timebase_tim.c, 44

HAL_ResumeTick
 stm32f4xx_hal_timebase_tim.c, 45

HAL_SuspendTick
 stm32f4xx_hal_timebase_tim.c, 45

HAL_TIM_PeriodElapsedCallback
 main.c, 43

HAL_UART_RxCpltCallback
 console.c, 36

heartBeatInit
 bsp.c, 19

HSE_VALUE
 STM32F4xx_System_Private_Includes, 8

HSI_VALUE
 STM32F4xx_System_Private_Includes, 8

main.c
 Error_Handler, 43
 HAL_TIM_PeriodElapsedCallback, 43

main.h
 Error_Handler, 26

prvCommandClk
 console.c, 36

prvCommandEcho
 console.c, 36

prvCommandGpioRead
 console.c, 37

prvCommandGpioWrite
 console.c, 37

prvCommandHeap
 console.c, 38

prvCommandPwmSetDuty
 console.c, 38

prvCommandPwmSetFreq
 console.c, 39

prvCommandTaskStats
 console.c, 39

prvCommandTicks
 console.c, 39

prvpcMapTaskState
 console.c, 40

prvpcTaskListHeader
 console.c, 42

Pwm_TIM_OC_InitTypeDef, 13

PwmConfigStruct, 13

stm32f4xx_hal_timebase_tim.c
 HAL_InitTick, 44
 HAL_ResumeTick, 45
 HAL_SuspendTick, 45

Stm32f4xx_system, 7

STM32F4xx_System_Private_Defines, 8

STM32F4xx_System_Private_FunctionPrototypes, 8

STM32F4xx_System_Private_Functions, 8
 SystemCoreClockUpdate, 9
 SystemInit, 9

STM32F4xx_System_Private_Includes, 7
 HSE_VALUE, 8
 HSI_VALUE, 8

STM32F4xx_System_Private_Macros, 8

STM32F4xx_System_Private_TypesDefinitions, 8

STM32F4xx_System_Private_Variables, 8

sysmem.c
 __sbrk_heap_end, 48
 _sbrk, 47

SystemCoreClockUpdate
 STM32F4xx_System_Private_Functions, 9

SystemInit
 STM32F4xx_System_Private_Functions, 9

vConsoleEnableRxInterrupt
 console.c, 40

vConsoleWrite
 console.c, 40

vTaskConsole
 console.c, 41

xConsoleInit
 console.c, 41

xConsoleRead
 console.c, 41