

PMIK

Generated by Doxygen 1.8.17

1 Module Index	1
1.1 Modules	1
2 File Index	3
2.1 File List	3
3 Module Documentation	5
3.1 CMSIS	5
3.1.1 Detailed Description	5
3.2 Stm32L1xx_system	6
3.2.1 Detailed Description	6
3.3 STM32L1xx_System_Private_Includes	7
3.4 STM32L1xx_System_Private_TypesDefinitions	8
3.5 STM32L1xx_System_Private_Defines	9
3.5.1 Detailed Description	9
3.5.2 Macro Definition Documentation	9
3.5.2.1 HSE_VALUE	9
3.5.2.2 HSI_VALUE	9
3.5.2.3 VECT_TAB_OFFSET	9
3.6 STM32L1xx_System_Private_Macros	10
3.7 STM32L1xx_System_Private_Variables	11
3.7.1 Detailed Description	11
3.7.2 Variable Documentation	11
3.7.2.1 AHBPRescTable	11
3.7.2.2 APBPRescTable	11
3.7.2.3 PLLMulTable	11
3.7.2.4 SystemCoreClock	11
3.8 STM32L1xx_System_Private_FunctionPrototypes	12
3.9 STM32L1xx_System_Private_Functions	13
3.9.1 Detailed Description	13
3.9.2 Function Documentation	13
3.9.2.1 SystemCoreClockUpdate()	13
3.9.2.2 SystemInit()	14
4 File Documentation	15
4.1 adc.c File Reference	15
4.1.1 Function Documentation	15
4.1.1.1 HAL_ADC_MspDeInit()	15
4.1.1.2 HAL_ADC_MspInit()	15
4.1.1.3 MX_ADC_Init()	16
4.1.2 Variable Documentation	16
4.1.2.1 hadc	16
4.2 adc.h File Reference	16

4.2.1 Function Documentation	17
4.2.1.1 MX_ADC_Init()	17
4.2.2 Variable Documentation	17
4.2.2.1 hadc	17
4.3 fotodetector.c File Reference	18
4.3.1 Function Documentation	18
4.3.1.1 get_value_fotodetector()	18
4.4 fotodetector.h File Reference	18
4.4.1 Function Documentation	18
4.4.1.1 get_value_fotodetector()	18
4.4.2 Variable Documentation	19
4.4.2.1 fotodetector_value	19
4.5 gpio.c File Reference	19
4.5.1 Function Documentation	19
4.5.1.1 MX_GPIO_Init()	19
4.6 gpio.h File Reference	20
4.6.1 Function Documentation	20
4.6.1.1 MX_GPIO_Init()	20
4.7 main.c File Reference	21
4.7.1 Detailed Description	21
4.7.2 Function Documentation	21
4.7.2.1 Error_Handler()	21
4.7.2.2 HAL_UART_RxCpltCallback()	22
4.7.2.3 main()	22
4.7.2.4 SystemClock_Config()	22
4.8 main.h File Reference	23
4.8.1 Detailed Description	24
4.8.2 Macro Definition Documentation	24
4.8.2.1 B1_EXTI_IRQn	24
4.8.2.2 B1_GPIO_Port	24
4.8.2.3 B1_Pin	24
4.8.2.4 DIR_GPIO_Port	24
4.8.2.5 DIR_Pin	25
4.8.2.6 ENABLE_GPIO_Port	25
4.8.2.7 ENABLE_Pin	25
4.8.2.8 END_DOWN_EXTI_IRQn	25
4.8.2.9 END_DOWN_GPIO_Port	25
4.8.2.10 END_DOWN_Pin	25
4.8.2.11 END_HIGH_EXTI_IRQn	25
4.8.2.12 END_HIGH_GPIO_Port	25
4.8.2.13 END_HIGH_Pin	26
4.8.2.14 FOTODETECTOR_GPIO_Port	26

4.8.2.15 FOTODETECTOR_Pin	26
4.8.2.16 LD2_GPIO_Port	26
4.8.2.17 LD2_Pin	26
4.8.2.18 STEP_GPIO_Port	26
4.8.2.19 STEP_Pin	26
4.8.2.20 SWO_GPIO_Port	26
4.8.2.21 SWO_Pin	27
4.8.2.22 TCK_GPIO_Port	27
4.8.2.23 TCK_Pin	27
4.8.2.24 TMS_GPIO_Port	27
4.8.2.25 TMS_Pin	27
4.8.2.26 USART_RX_GPIO_Port	27
4.8.2.27 USART_RX_Pin	27
4.8.2.28 USART_TX_GPIO_Port	27
4.8.2.29 USART_TX_Pin	28
4.8.3 Function Documentation	28
4.8.3.1 Error_Handler()	28
4.8.4 Variable Documentation	28
4.8.4.1 able	28
4.8.4.2 data	28
4.9 motor.c File Reference	28
4.9.1 Function Documentation	29
4.9.1.1 admin_pc()	29
4.9.1.2 auto_fotodetector()	29
4.9.1.3 HAL_GPIO_EXTI_Callback()	29
4.9.1.4 home_down()	29
4.9.1.5 home_up()	29
4.9.1.6 move_motor()	30
4.9.1.7 new_data()	30
4.10 motor.h File Reference	30
4.10.1 Function Documentation	30
4.10.1.1 admin_pc()	30
4.10.1.2 auto_fotodetector()	31
4.10.1.3 home_down()	31
4.10.1.4 home_up()	31
4.10.1.5 move_motor()	31
4.10.1.6 new_data()	31
4.10.2 Variable Documentation	31
4.10.2.1 admin	31
4.10.2.2 auto_down	31
4.10.2.3 auto_up	32
4.10.2.4 control	32

4.10.2.5 count_position	32
4.10.2.6 dir	32
4.10.2.7 position	32
4.10.2.8 set_position	32
4.11 stm32l1xx_hal_conf.h File Reference	32
4.11.1 Detailed Description	34
4.11.2 Macro Definition Documentation	34
4.11.2.1 assert_param	34
4.11.2.2 DATA_CACHE_ENABLE	35
4.11.2.3 HAL_ADC_MODULE_ENABLED	35
4.11.2.4 HAL_CORTEX_MODULE_ENABLED	35
4.11.2.5 HAL_DMA_MODULE_ENABLED	35
4.11.2.6 HAL_FLASH_MODULE_ENABLED	35
4.11.2.7 HAL_GPIO_MODULE_ENABLED	35
4.11.2.8 HAL_MODULE_ENABLED	35
4.11.2.9 HAL_PWR_MODULE_ENABLED	36
4.11.2.10 HAL_RCC_MODULE_ENABLED	36
4.11.2.11 HAL_UART_MODULE_ENABLED	36
4.11.2.12 HSE_STARTUP_TIMEOUT	36
4.11.2.13 HSE_VALUE	36
4.11.2.14 HSI_VALUE	36
4.11.2.15 INSTRUCTION_CACHE_ENABLE	36
4.11.2.16 LSE_STARTUP_TIMEOUT	37
4.11.2.17 LSE_VALUE	37
4.11.2.18 LSI_VALUE	37
4.11.2.19 MSI_VALUE	37
4.11.2.20 PREFETCH_ENABLE	37
4.11.2.21 TICK_INT_PRIORITY	37
4.11.2.22 USE_HAL_ADC_REGISTER_CALLBACKS	38
4.11.2.23 USE_HAL_COMP_REGISTER_CALLBACKS	38
4.11.2.24 USE_HAL_DAC_REGISTER_CALLBACKS	38
4.11.2.25 USE_HAL_I2C_REGISTER_CALLBACKS	38
4.11.2.26 USE_HAL_I2S_REGISTER_CALLBACKS	38
4.11.2.27 USE_HAL_IRDA_REGISTER_CALLBACKS	38
4.11.2.28 USE_HAL_OPAMP_REGISTER_CALLBACKS	38
4.11.2.29 USE_HAL_PCD_REGISTER_CALLBACKS	39
4.11.2.30 USE_HAL_RTC_REGISTER_CALLBACKS	39
4.11.2.31 USE_HAL_SDMMC_REGISTER_CALLBACKS	39
4.11.2.32 USE_HAL_SMARTCARD_REGISTER_CALLBACKS	39
4.11.2.33 USE_HAL_SPI_REGISTER_CALLBACKS	39
4.11.2.34 USE_HAL_TIM_REGISTER_CALLBACKS	39
4.11.2.35 USE_HAL_UART_REGISTER_CALLBACKS	39

4.11.2.36 USE_HAL_USART_REGISTER_CALLBACKS	39
4.11.2.37 USE_HAL_WWDG_REGISTER_CALLBACKS	40
4.11.2.38 USE_RTOS	40
4.11.2.39 USE_SPI_CRC	40
4.11.2.40 VDD_VALUE	40
4.12 stm32l1xx_hal_msp.c File Reference	40
4.12.1 Function Documentation	40
4.12.1.1 HAL_MspInit()	40
4.13 stm32l1xx_it.c File Reference	41
4.13.1 Detailed Description	42
4.13.2 Function Documentation	42
4.13.2.1 ADC1_IRQHandler()	42
4.13.2.2 BusFault_Handler()	42
4.13.2.3 DebugMon_Handler()	43
4.13.2.4 EXTI15_10_IRQHandler()	43
4.13.2.5 EXTI9_5_IRQHandler()	43
4.13.2.6 HardFault_Handler()	43
4.13.2.7 MemManage_Handler()	43
4.13.2.8 NMI_Handler()	43
4.13.2.9 PendSV_Handler()	44
4.13.2.10 SVC_Handler()	44
4.13.2.11 SysTick_Handler()	44
4.13.2.12 UsageFault_Handler()	44
4.13.2.13 USART1_IRQHandler()	44
4.13.2.14 USART2_IRQHandler()	44
4.13.2.15 USART3_IRQHandler()	45
4.13.3 Variable Documentation	45
4.13.3.1 hadc	45
4.13.3.2 huart1	45
4.13.3.3 huart2	46
4.13.3.4 huart3	46
4.14 stm32l1xx_it.h File Reference	46
4.14.1 Detailed Description	47
4.14.2 Function Documentation	47
4.14.2.1 ADC1_IRQHandler()	47
4.14.2.2 BusFault_Handler()	47
4.14.2.3 DebugMon_Handler()	47
4.14.2.4 EXTI15_10_IRQHandler()	48
4.14.2.5 EXTI9_5_IRQHandler()	48
4.14.2.6 HardFault_Handler()	48
4.14.2.7 MemManage_Handler()	48
4.14.2.8 NMI_Handler()	48

4.14.2.9 PendSV_Handler()	48
4.14.2.10 SVC_Handler()	49
4.14.2.11 SysTick_Handler()	49
4.14.2.12 UsageFault_Handler()	49
4.14.2.13 USART1_IRQHandler()	49
4.14.2.14 USART2_IRQHandler()	49
4.14.2.15 USART3_IRQHandler()	49
4.15 syscalls.c File Reference	50
4.15.1 Detailed Description	50
4.15.2 Function Documentation	51
4.15.2.1 __attribute__()	51
4.15.2.2 __io_getchar()	51
4.15.2.3 __io_putchar()	51
4.15.2.4 _close()	51
4.15.2.5 _execve()	51
4.15.2.6 _exit()	52
4.15.2.7 _fork()	52
4.15.2.8 _fstat()	52
4.15.2.9 _getpid()	52
4.15.2.10 _isatty()	52
4.15.2.11 _kill()	52
4.15.2.12 _link()	53
4.15.2.13 _lseek()	53
4.15.2.14 _open()	53
4.15.2.15 _stat()	53
4.15.2.16 _times()	53
4.15.2.17 _unlink()	53
4.15.2.18 _wait()	54
4.15.2.19 initialise_monitor_handles()	54
4.15.3 Variable Documentation	54
4.15.3.1 environ	54
4.15.3.2 errno	54
4.16 systemmem.c File Reference	54
4.16.1 Detailed Description	55
4.16.2 Function Documentation	55
4.16.2.1 _sbrk()	55
4.16.2.2 asm()	55
4.16.3 Variable Documentation	55
4.16.3.1 errno	55
4.17 system_stm32l1xx.c File Reference	56
4.17.1 Detailed Description	56
4.18 usart.c File Reference	57

4.18.1 Function Documentation	57
4.18.1.1 HAL_UART_MspDeInit()	57
4.18.1.2 HAL_UART_MspInit()	58
4.18.1.3 MX_USART1_UART_Init()	58
4.18.1.4 MX_USART2_UART_Init()	58
4.18.1.5 MX_USART3_UART_Init()	58
4.18.2 Variable Documentation	58
4.18.2.1 huart1	58
4.18.2.2 huart2	59
4.18.2.3 huart3	59
4.19 usart.h File Reference	59
4.19.1 Function Documentation	59
4.19.1.1 MX_USART1_UART_Init()	59
4.19.1.2 MX_USART2_UART_Init()	60
4.19.1.3 MX_USART3_UART_Init()	60
4.19.2 Variable Documentation	60
4.19.2.1 huart1	60
4.19.2.2 huart2	60
4.19.2.3 huart3	60
Index	61

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

CMSIS	5
Stm32l1xx_system	6
STM32L1xx_System_Private_Includes	7
STM32L1xx_System_Private_TypesDefinitions	8
STM32L1xx_System_Private_Defines	9
STM32L1xx_System_Private_Macros	10
STM32L1xx_System_Private_Variables	11
STM32L1xx_System_Private_FunctionPrototypes	12
STM32L1xx_System_Private_Functions	13

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

adc.c	15
adc.h	16
fotodetector.c	18
fotodetector.h	18
gpio.c	19
gpio.h	20
main.c	
: Main program body	21
main.h	
: Header for main.c file. This file contains the common defines of the application	23
motor.c	28
motor.h	30
stm32l1xx_hal_conf.h	
HAL configuration file.	
32	
stm32l1xx_hal_msp.c	40
stm32l1xx_it.c	
Interrupt Service Routines	41
stm32l1xx_it.h	
This file contains the headers of the interrupt handlers	46
syscalls.c	
STM32CubeIDE Minimal System calls file	50
sysmem.c	
STM32CubeIDE Minimal System Memory calls file	54
system_stm32l1xx.c	
CMSIS Cortex-M3 Device Peripheral Access Layer System Source File	56
usart.c	57
usart.h	59

Chapter 3

Module Documentation

3.1 CMSIS

Modules

- [Stm32l1xx_system](#)

3.1.1 Detailed Description

3.2 Stm32l1xx_system

Modules

- [STM32L1xx_System_Private_Includes](#)
- [STM32L1xx_System_Private_TypesDefinitions](#)
- [STM32L1xx_System_Private_Defines](#)
- [STM32L1xx_System_Private_Macros](#)
- [STM32L1xx_System_Private_Variables](#)
- [STM32L1xx_System_Private_FunctionPrototypes](#)
- [STM32L1xx_System_Private_Functions](#)

3.2.1 Detailed Description

3.3 STM32L1xx_System_Private_Includes

3.4 STM32L1xx_System_Private_TypesDefinitions

3.5 STM32L1xx_System_Private_Defines

Macros

- #define `HSE_VALUE` ((uint32_t)8000000U)
- #define `HSI_VALUE` ((uint32_t)8000000U)
- #define `VECT_TAB_OFFSET` 0x00U

3.5.1 Detailed Description

3.5.2 Macro Definition Documentation

3.5.2.1 HSE_VALUE

```
#define HSE_VALUE ((uint32_t)8000000U)
```

Default value of the External oscillator in Hz. This value can be provided and adapted by the user application.

3.5.2.2 HSI_VALUE

```
#define HSI_VALUE ((uint32_t)8000000U)
```

Default value of the Internal oscillator in Hz. This value can be provided and adapted by the user application.

3.5.2.3 VECT_TAB_OFFSET

```
#define VECT_TAB_OFFSET 0x00U
```

< Uncomment the following line if you need to use external SRAM mounted on STM32L152D_EVAL board 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.

3.6 STM32L1xx_System_Private_Macros

3.7 STM32L1xx_System_Private_Variables

Variables

- uint32_t [SystemCoreClock](#) = 2097000U
- const uint8_t [PLLMulTable](#) [9] = {3U, 4U, 6U, 8U, 12U, 16U, 24U, 32U, 48U}
- const uint8_t [AHBPrescTable](#) [16] = {0U, 0U, 0U, 0U, 0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U, 6U, 7U, 8U, 9U}
- const uint8_t [APBPrescTable](#) [8] = {0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U}

3.7.1 Detailed Description

3.7.2 Variable Documentation

3.7.2.1 AHBPrescTable

```
const uint8_t AHBPrescTable[16] = {0U, 0U, 0U, 0U, 0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U, 6U, 7U, 8U, 9U}
```

3.7.2.2 APBPrescTable

```
const uint8_t APBPrescTable[8] = {0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U}
```

3.7.2.3 PLLMulTable

```
const uint8_t PLLMulTable[9] = {3U, 4U, 6U, 8U, 12U, 16U, 24U, 32U, 48U}
```

3.7.2.4 SystemCoreClock

```
uint32_t SystemCoreClock = 2097000U
```

3.8 STM32L1xx_System_Private_FunctionPrototypes

3.9 STM32L1xx_System_Private_Functions

Functions

- void [SystemInit](#) (void)
Setup the microcontroller system. Initialize the Embedded Flash Interface, the PLL and update the SystemCoreClock variable.
- void [SystemCoreClockUpdate](#) (void)
Update SystemCoreClock 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.

3.9.1 Detailed Description

3.9.2 Function Documentation

3.9.2.1 SystemCoreClockUpdate()

```
void SystemCoreClockUpdate (
    void )
```

Update SystemCoreClock 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 MSI, SystemCoreClock will contain the MSI value as defined by the MSI range.
- 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 stm32l1xx.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 stm32l1xx.h file (default value 8 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	
------	--

3.9.2.2 SystemInit()

```
void SystemInit (
    void )
```

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

Parameters

None	
------	--

Return values

None	
------	--

< Set MSION bit

< Reset SW[1:0], HPRE[3:0], PPRE1[2:0], PPRE2[2:0], MCOSEL[2:0] and MCOPRE[2:0] bits

< Reset HSION, HSEON, CSSON and PLLON bits

< Reset HSEBYP bit

< Reset PLLSRC, PLLMUL[3:0] and PLLDIV[1:0] bits

< Disable all interrupts

Chapter 4

File Documentation

4.1 adc.c File Reference

```
#include "adc.h"
```

Functions

- void [MX_ADC_Init](#) (void)
- void [HAL_ADC_MspInit](#) (ADC_HandleTypeDef *adcHandle)
- void [HAL_ADC_MspDeInit](#) (ADC_HandleTypeDef *adcHandle)

Variables

- ADC_HandleTypeDef [hadc](#)

4.1.1 Function Documentation

4.1.1.1 HAL_ADC_MspDeInit()

```
void HAL_ADC_MspDeInit (
    ADC_HandleTypeDef * adcHandle )
```

ADC GPIO Configuration
PA0-WKUP1 ----> ADC_IN0

4.1.1.2 HAL_ADC_MspInit()

```
void HAL_ADC_MspInit (
    ADC_HandleTypeDef * adcHandle )
```

ADC GPIO Configuration
PA0-WKUP1 ----> ADC_IN0

4.1.1.3 MX_ADC_Init()

```
void MX_ADC_Init (
    void )
```

Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of conversion)

Configure for the selected ADC regular channel its corresponding rank in the sequencer and its sample time.

4.1.2 Variable Documentation

4.1.2.1 hadc

```
ADC_HandleTypeDef hadc
```

File Name : [ADC.c](#) Description : This file provides code for the configuration of the ADC instances.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.2 adc.h File Reference

```
#include "main.h"
```

Functions

- void [MX_ADC_Init](#) (void)

Variables

- ADC_HandleTypeDef [hadc](#)

4.2.1 Function Documentation

4.2.1.1 MX_ADC_Init()

```
void MX_ADC_Init (
    void )
```

Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of conversion)

Configure for the selected ADC regular channel its corresponding rank in the sequencer and its sample time.

4.2.2 Variable Documentation

4.2.2.1 hadc

```
ADC_HandleTypeDef hadc
```

File Name : [ADC.h](#) Description : This file provides code for the configuration of the ADC instances.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

File Name : [ADC.c](#) Description : This file provides code for the configuration of the ADC instances.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.3 fotodetector.c File Reference

```
#include "fotodetector.h"
```

Functions

- `uint8_t get_value_fotodetector ()`

4.3.1 Function Documentation

4.3.1.1 get_value_fotodetector()

```
uint8_t get_value_fotodetector ( )
```

File Name : [fotodetector.c](#) Description : This file provides code for the configuration of the fotodetector.

4.4 fotodetector.h File Reference

```
#include "main.h"  
#include "adc.h"  
#include "usart.h"
```

Functions

- `uint8_t get_value_fotodetector ()`

Variables

- `uint8_t fotodetector_value`

4.4.1 Function Documentation

4.4.1.1 get_value_fotodetector()

```
uint8_t get_value_fotodetector ( )
```

function for get valude from fotodetector

File Name : [fotodetector.c](#) Description : This file provides code for the configuration of the fotodetector.

4.4.2 Variable Documentation

4.4.2.1 fotodetector_value

```
uint8_t fotodetector_value
```

File Name : [fotodetector.h](#) Description : This file provides code for the configuration of the fotodetector value taken from fotodetector

4.5 gpio.c File Reference

```
#include "gpio.h"
```

Functions

- void [MX_GPIO_Init](#) (void)

4.5.1 Function Documentation

4.5.1.1 MX_GPIO_Init()

```
void MX_GPIO_Init (  
    void )
```

File Name : [gpio.c](#) Description : This file provides code for the configuration of all used GPIO pins.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause Configure pins as Analog Input Output EVENT_OUT EXTI

4.6 gpio.h File Reference

```
#include "main.h"
```

Functions

- void [MX_GPIO_Init](#) (void)

4.6.1 Function Documentation

4.6.1.1 MX_GPIO_Init()

```
void MX_GPIO_Init (  
    void )
```

File Name : [gpio.h](#) Description : This file contains all the functions prototypes for the gpio

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

File Name : [gpio.c](#) Description : This file provides code for the configuration of all used GPIO pins.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause Configure pins as Analog Input Output EVENT_OUT EXTI

4.7 main.c File Reference

: Main program body

```
#include "main.h"
#include "adc.h"
#include "usart.h"
#include "gpio.h"
```

Functions

- void [SystemClock_Config](#) (void)
System Clock Configuration.
- int [main](#) (void)
The application entry point.
- void [HAL_UART_RxCpltCallback](#) (UART_HandleTypeDef *huart)
- void [Error_Handler](#) (void)
This function is executed in case of error occurrence.

4.7.1 Detailed Description

: Main program body

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.7.2 Function Documentation

4.7.2.1 Error_Handler()

```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

Return values

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

4.7.2.2 HAL_UART_RxCpltCallback()

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

uart take command raspberry uart

admin uart

4.7.2.3 main()

```
int main (
    void )
```

The application entry point.

Return values

<i>int</i>	
------------	--

wait for uart command from raspberry huart1 is in exit mode

start adc read

wait for uart command from admin huart2 is in exit mode

4.7.2.4 SystemClock_Config()

```
void SystemClock_Config (
    void )
```

System Clock Configuration.

Return values

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

Configure the main internal regulator output voltage

Initializes the CPU, AHB and APB busses clocks

Initializes the CPU, AHB and APB busses clocks

4.8 main.h File Reference

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

```
#include "stm32l1xx_hal.h"
```

Macros

- `#define B1_Pin` GPIO_PIN_13
- `#define B1_GPIO_Port` GPIOC
- `#define B1_EXTI_IRQn` EXTI15_10_IRQn
- `#define FOTODETECTOR_Pin` GPIO_PIN_0
- `#define FOTODETECTOR_GPIO_Port` GPIOA
- `#define USART_TX_Pin` GPIO_PIN_2
- `#define USART_TX_GPIO_Port` GPIOA
- `#define USART_RX_Pin` GPIO_PIN_3
- `#define USART_RX_GPIO_Port` GPIOA
- `#define LD2_Pin` GPIO_PIN_5
- `#define LD2_GPIO_Port` GPIOA
- `#define ENABLE_Pin` GPIO_PIN_15
- `#define ENABLE_GPIO_Port` GPIOB
- `#define END_DOWN_Pin` GPIO_PIN_6
- `#define END_DOWN_GPIO_Port` GPIOC
- `#define END_DOWN_EXTI_IRQn` EXTI9_5_IRQn
- `#define STEP_Pin` GPIO_PIN_7
- `#define STEP_GPIO_Port` GPIOC
- `#define DIR_Pin` GPIO_PIN_8
- `#define DIR_GPIO_Port` GPIOC
- `#define END_HIGH_Pin` GPIO_PIN_9
- `#define END_HIGH_GPIO_Port` GPIOC
- `#define END_HIGH_EXTI_IRQn` EXTI9_5_IRQn
- `#define TMS_Pin` GPIO_PIN_13
- `#define TMS_GPIO_Port` GPIOA
- `#define TCK_Pin` GPIO_PIN_14
- `#define TCK_GPIO_Port` GPIOA
- `#define SWO_Pin` GPIO_PIN_3
- `#define SWO_GPIO_Port` GPIOB

Functions

- void [Error_Handler](#) (void)

This function is executed in case of error occurrence.

Variables

- uint8_t [data](#)
- uint8_t [able](#)

4.8.1 Detailed Description

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

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.8.2 Macro Definition Documentation

4.8.2.1 B1_EXTI_IRQn

```
#define B1_EXTI_IRQn EXTI15_10_IRQn
```

4.8.2.2 B1_GPIO_Port

```
#define B1_GPIO_Port GPIOC
```

4.8.2.3 B1_Pin

```
#define B1_Pin GPIO_PIN_13
```

4.8.2.4 DIR_GPIO_Port

```
#define DIR_GPIO_Port GPIOC
```

4.8.2.5 DIR_Pin

```
#define DIR_Pin GPIO_PIN_8
```

4.8.2.6 ENABLE_GPIO_Port

```
#define ENABLE_GPIO_Port GPIOB
```

4.8.2.7 ENABLE_Pin

```
#define ENABLE_Pin GPIO_PIN_15
```

4.8.2.8 END_DOWN_EXTI_IRQn

```
#define END_DOWN_EXTI_IRQn EXTI9_5_IRQn
```

4.8.2.9 END_DOWN_GPIO_Port

```
#define END_DOWN_GPIO_Port GPIOC
```

4.8.2.10 END_DOWN_Pin

```
#define END_DOWN_Pin GPIO_PIN_6
```

4.8.2.11 END_HIGH_EXTI_IRQn

```
#define END_HIGH_EXTI_IRQn EXTI9_5_IRQn
```

4.8.2.12 END_HIGH_GPIO_Port

```
#define END_HIGH_GPIO_Port GPIOC
```

4.8.2.13 END_HIGH_Pin

```
#define END_HIGH_Pin GPIO_PIN_9
```

4.8.2.14 FOTODETECTOR_GPIO_Port

```
#define FOTODETECTOR_GPIO_Port GPIOA
```

4.8.2.15 FOTODETECTOR_Pin

```
#define FOTODETECTOR_Pin GPIO_PIN_0
```

4.8.2.16 LD2_GPIO_Port

```
#define LD2_GPIO_Port GPIOA
```

4.8.2.17 LD2_Pin

```
#define LD2_Pin GPIO_PIN_5
```

4.8.2.18 STEP_GPIO_Port

```
#define STEP_GPIO_Port GPIOC
```

4.8.2.19 STEP_Pin

```
#define STEP_Pin GPIO_PIN_7
```

4.8.2.20 SWO_GPIO_Port

```
#define SWO_GPIO_Port GPIOB
```

4.8.2.21 SWO_Pin

```
#define SWO_Pin GPIO_PIN_3
```

4.8.2.22 TCK_GPIO_Port

```
#define TCK_GPIO_Port GPIOA
```

4.8.2.23 TCK_Pin

```
#define TCK_Pin GPIO_PIN_14
```

4.8.2.24 TMS_GPIO_Port

```
#define TMS_GPIO_Port GPIOA
```

4.8.2.25 TMS_Pin

```
#define TMS_Pin GPIO_PIN_13
```

4.8.2.26 USART_RX_GPIO_Port

```
#define USART_RX_GPIO_Port GPIOA
```

4.8.2.27 USART_RX_Pin

```
#define USART_RX_Pin GPIO_PIN_3
```

4.8.2.28 USART_TX_GPIO_Port

```
#define USART_TX_GPIO_Port GPIOA
```

4.8.2.29 USART_TX_Pin

```
#define USART_TX_Pin GPIO_PIN_2
```

4.8.3 Function Documentation

4.8.3.1 Error_Handler()

```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

Return values

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

4.8.4 Variable Documentation

4.8.4.1 able

```
uint8_t able
```

data from admin

4.8.4.2 data

```
uint8_t data
```

data from raspberry

4.9 motor.c File Reference

```
#include "motor.h"
```

Functions

- void [HAL_GPIO_EXTI_Callback](#) (uint16_t GPIO_Pin)
- void [new_data](#) (uint8_t new)
- void [admin_pc](#) (uint8_t *able*)
- void [move_motor](#) ()
- void [home_up](#) ()
- void [home_down](#) ()
- void [auto_fotodetector](#) (uint8_t auto_foto)

4.9.1 Function Documentation

4.9.1.1 admin_pc()

```
void admin_pc (  
    uint8_t able )
```

work permit

4.9.1.2 auto_fotodetector()

```
void auto_fotodetector (  
    uint8_t auto_foto )
```

function to execution comand from fotodetector

4.9.1.3 HAL_GPIO_EXTI_Callback()

```
void HAL_GPIO_EXTI_Callback (  
    uint16_t GPIO_Pin )
```

File Name : [motor.c](#) Description : This file provides code for the configuration of the motor. function for execute end command

4.9.1.4 home_down()

```
void home_down ( )
```

homeing down blind function

4.9.1.5 home_up()

```
void home_up ( )
```

homeing up blind function

4.9.1.6 move_motor()

```
void move_motor ( )
```

function that sends commands to the engine

4.9.1.7 new_data()

```
void new_data (
    uint8_t new )
```

function to execution comand from uart

4.10 motor.h File Reference

```
#include "main.h"
```

Functions

- void [move_motor](#) ()
- void [new_data](#) (uint8_t new)
- void [home_up](#) ()
- void [home_down](#) ()
- void [auto_fotodetector](#) (uint8_t auto_foto)
- void [admin_pc](#) (uint8_t [able](#))

Variables

- uint8_t [set_position](#)
- uint8_t [position](#)
- uint8_t [control](#)
- int [dir](#)
- int [count_position](#) = 0
- int [auto_down](#)
- int [auto_up](#)
- uint8_t [admin](#) = 1

4.10.1 Function Documentation

4.10.1.1 admin_pc()

```
void admin_pc (
    uint8_t able )
```

work permit

4.10.1.2 auto_fotodetector()

```
void auto_fotodetector (
    uint8_t auto_foto )
```

function to execution comand from fotodetector

4.10.1.3 home_down()

```
void home_down ( )
```

homeing down blind function

4.10.1.4 home_up()

```
void home_up ( )
```

homeing up blind function

4.10.1.5 move_motor()

```
void move_motor ( )
```

function that sends commands to the engine

4.10.1.6 new_data()

```
void new_data (
    uint8_t new )
```

function to execution comand from uart

4.10.2 Variable Documentation

4.10.2.1 admin

```
uint8_t admin = 1
```

work permit

4.10.2.2 auto_down

```
int auto_down
```

motor can go down in auto mode

4.10.2.3 auto_up

```
int auto_up
```

motor can go up in auto mode

4.10.2.4 control

```
uint8_t control
```

value to choose auto mode

4.10.2.5 count_position

```
int count_position = 0
```

helper variable to include the number of steps in one iteration

4.10.2.6 dir

```
int dir
```

4.10.2.7 position

```
uint8_t position
```

actual position

4.10.2.8 set_position

```
uint8_t set_position
```

File Name : [motor.h](#) Description : This file provides code for the configuration of the motor value for setting position

4.11 stm32l1xx_hal_conf.h File Reference

HAL configuration file.

```
#include "stm32l1xx_hal_rcc.h"
#include "stm32l1xx_hal_gpio.h"
#include "stm32l1xx_hal_dma.h"
#include "stm32l1xx_hal_cortex.h"
#include "stm32l1xx_hal_adc.h"
#include "stm32l1xx_hal_flash.h"
#include "stm32l1xx_hal_pwr.h"
#include "stm32l1xx_hal_uart.h"
```

Macros

- #define `HAL_MODULE_ENABLED`
This is the list of modules to be used in the HAL driver.
- #define `HAL_ADC_MODULE_ENABLED`
- #define `HAL_UART_MODULE_ENABLED`
- #define `HAL_GPIO_MODULE_ENABLED`
- #define `HAL_DMA_MODULE_ENABLED`
- #define `HAL_RCC_MODULE_ENABLED`
- #define `HAL_FLASH_MODULE_ENABLED`
- #define `HAL_PWR_MODULE_ENABLED`
- #define `HAL_CORTEX_MODULE_ENABLED`
- #define `HSE_VALUE` ((uint32_t)8000000)
Adjust the value of External High Speed oscillator (HSE) used in your application. This value is used by the RCC HAL module to compute the system frequency (when HSE is used as system clock source, directly or through the PLL).
- #define `HSE_STARTUP_TIMEOUT` ((uint32_t)100)
- #define `MSI_VALUE` ((uint32_t)16000000)
Internal Multiple Speed oscillator (MSI) default value. This value is the default MSI range value after Reset.
- #define `HSI_VALUE` ((uint32_t)16000000)
Internal High Speed oscillator (HSI) value. This value is used by the RCC HAL module to compute the system frequency (when HSI is used as system clock source, directly or through the PLL).
- #define `LSI_VALUE` (37000U)
Internal Low Speed oscillator (LSI) value.
- #define `LSE_VALUE` ((uint32_t)32768)
External Low Speed oscillator (LSE) value. This value is used by the UART, RTC HAL module to compute the system frequency.
- #define `LSE_STARTUP_TIMEOUT` ((uint32_t)5000)
- #define `VDD_VALUE` ((uint32_t)3300)
This is the HAL system configuration section.
- #define `TICK_INT_PRIORITY` ((uint32_t)0)
- #define `USE_RTOS` 0
- #define `PREFETCH_ENABLE` 0
- #define `INSTRUCTION_CACHE_ENABLE` 1
- #define `DATA_CACHE_ENABLE` 1
- #define `USE_HAL_ADC_REGISTER_CALLBACKS` 0U
Uncomment the line below to expanse the "assert_param" macro in the HAL drivers code.
- #define `USE_HAL_COMP_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_DAC_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_I2C_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_I2S_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_IRDA_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_OPAMP_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_PCD_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_RTC_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_SDMMC_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_SMARTCARD_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_SPI_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_TIM_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_UART_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_USART_REGISTER_CALLBACKS` 0U
- #define `USE_HAL_WWDG_REGISTER_CALLBACKS` 0U
- #define `USE_SPI_CRC` 0U
- #define `assert_param`(expr) ((void)0U)
Include module's header file.

4.11.1 Detailed Description

HAL configuration file.

Attention

© COPYRIGHT(c) 2020 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

4.11.2 Macro Definition Documentation

4.11.2.1 `assert_param`

```
#define assert_param(  
    expr ) ((void)0U)
```

Include module's header file.

4.11.2.2 DATA_CACHE_ENABLE

```
#define DATA_CACHE_ENABLE 1
```

4.11.2.3 HAL_ADC_MODULE_ENABLED

```
#define HAL_ADC_MODULE_ENABLED
```

4.11.2.4 HAL_CORTEX_MODULE_ENABLED

```
#define HAL_CORTEX_MODULE_ENABLED
```

4.11.2.5 HAL_DMA_MODULE_ENABLED

```
#define HAL_DMA_MODULE_ENABLED
```

4.11.2.6 HAL_FLASH_MODULE_ENABLED

```
#define HAL_FLASH_MODULE_ENABLED
```

4.11.2.7 HAL_GPIO_MODULE_ENABLED

```
#define HAL_GPIO_MODULE_ENABLED
```

4.11.2.8 HAL_MODULE_ENABLED

```
#define HAL_MODULE_ENABLED
```

This is the list of modules to be used in the HAL driver.

4.11.2.9 HAL_PWR_MODULE_ENABLED

```
#define HAL_PWR_MODULE_ENABLED
```

4.11.2.10 HAL_RCC_MODULE_ENABLED

```
#define HAL_RCC_MODULE_ENABLED
```

4.11.2.11 HAL_UART_MODULE_ENABLED

```
#define HAL_UART_MODULE_ENABLED
```

4.11.2.12 HSE_STARTUP_TIMEOUT

```
#define HSE_STARTUP_TIMEOUT ((uint32_t)100)
```

Time out for HSE start up, in ms

4.11.2.13 HSE_VALUE

```
#define HSE_VALUE ((uint32_t)8000000)
```

Adjust the value of External High Speed oscillator (HSE) used in your application. This value is used by the RCC HAL module to compute the system frequency (when HSE is used as system clock source, directly or through the PLL).

Value of the External oscillator in Hz

4.11.2.14 HSI_VALUE

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

Internal High Speed oscillator (HSI) value. This value is used by the RCC HAL module to compute the system frequency (when HSI is used as system clock source, directly or through the PLL).

Value of the Internal oscillator in Hz

4.11.2.15 INSTRUCTION_CACHE_ENABLE

```
#define INSTRUCTION_CACHE_ENABLE 1
```

4.11.2.16 LSE_STARTUP_TIMEOUT

```
#define LSE_STARTUP_TIMEOUT ((uint32_t)5000)
```

Time out for LSE start up, in ms

4.11.2.17 LSE_VALUE

```
#define LSE_VALUE ((uint32_t)32768)
```

External Low Speed oscillator (LSE) value. This value is used by the UART, RTC HAL module to compute the system frequency.

< Value of the Internal Low Speed oscillator in Hz The real value may vary depending on the variations in voltage and temperature. Value of the External oscillator in Hz

4.11.2.18 LSI_VALUE

```
#define LSI_VALUE (37000U)
```

Internal Low Speed oscillator (LSI) value.

LSI Typical Value in Hz

4.11.2.19 MSI_VALUE

```
#define MSI_VALUE ((uint32_t)16000000)
```

Internal Multiple Speed oscillator (MSI) default value. This value is the default MSI range value after Reset.

Value of the Internal oscillator in Hz

4.11.2.20 PREFETCH_ENABLE

```
#define PREFETCH_ENABLE 0
```

4.11.2.21 TICK_INT_PRIORITY

```
#define TICK_INT_PRIORITY ((uint32_t)0)
```

tick interrupt priority

4.11.2.22 USE_HAL_ADC_REGISTER_CALLBACKS

```
#define USE_HAL_ADC_REGISTER_CALLBACKS 0U
```

Uncomment the line below to expand the "assert_param" macro in the HAL drivers code.

Set below the peripheral configuration to "1U" to add the support of HAL callback registration/deregistration feature for the HAL driver(s). This allows user application to provide specific callback functions thanks to HAL_PPP_RegisterCallback() rather than overwriting the default weak callback functions (see each stm32l0xx_hal_ppp.h file for possible callback identifiers defined in HAL_PPP_CallbackIDTypeDef for each PPP peripheral).

4.11.2.23 USE_HAL_COMP_REGISTER_CALLBACKS

```
#define USE_HAL_COMP_REGISTER_CALLBACKS 0U
```

4.11.2.24 USE_HAL_DAC_REGISTER_CALLBACKS

```
#define USE_HAL_DAC_REGISTER_CALLBACKS 0U
```

4.11.2.25 USE_HAL_I2C_REGISTER_CALLBACKS

```
#define USE_HAL_I2C_REGISTER_CALLBACKS 0U
```

4.11.2.26 USE_HAL_I2S_REGISTER_CALLBACKS

```
#define USE_HAL_I2S_REGISTER_CALLBACKS 0U
```

4.11.2.27 USE_HAL_IRDA_REGISTER_CALLBACKS

```
#define USE_HAL_IRDA_REGISTER_CALLBACKS 0U
```

4.11.2.28 USE_HAL_OPAMP_REGISTER_CALLBACKS

```
#define USE_HAL_OPAMP_REGISTER_CALLBACKS 0U
```


4.11.2.29 USE_HAL_PCD_REGISTER_CALLBACKS

```
#define USE_HAL_PCD_REGISTER_CALLBACKS 0U
```

4.11.2.30 USE_HAL_RTC_REGISTER_CALLBACKS

```
#define USE_HAL_RTC_REGISTER_CALLBACKS 0U
```

4.11.2.31 USE_HAL_SDMMC_REGISTER_CALLBACKS

```
#define USE_HAL_SDMMC_REGISTER_CALLBACKS 0U
```

4.11.2.32 USE_HAL_SMARTCARD_REGISTER_CALLBACKS

```
#define USE_HAL_SMARTCARD_REGISTER_CALLBACKS 0U
```

4.11.2.33 USE_HAL_SPI_REGISTER_CALLBACKS

```
#define USE_HAL_SPI_REGISTER_CALLBACKS 0U
```

4.11.2.34 USE_HAL_TIM_REGISTER_CALLBACKS

```
#define USE_HAL_TIM_REGISTER_CALLBACKS 0U
```

4.11.2.35 USE_HAL_UART_REGISTER_CALLBACKS

```
#define USE_HAL_UART_REGISTER_CALLBACKS 0U
```

4.11.2.36 USE_HAL_USART_REGISTER_CALLBACKS

```
#define USE_HAL_USART_REGISTER_CALLBACKS 0U
```

4.11.2.37 USE_HAL_WWDG_REGISTER_CALLBACKS

```
#define USE_HAL_WWDG_REGISTER_CALLBACKS 0U
```

4.11.2.38 USE_RTOS

```
#define USE_RTOS 0
```

4.11.2.39 USE_SPI_CRC

```
#define USE_SPI_CRC 0U
```

4.11.2.40 VDD_VALUE

```
#define VDD_VALUE ((uint32_t)3300)
```

This is the HAL system configuration section.

Value of VDD in mv

4.12 stm32l1xx_hal_msp.c File Reference

```
#include "main.h"
```

Functions

- void [HAL_MspInit](#) (void)

4.12.1 Function Documentation

4.12.1.1 HAL_MspInit()

```
void HAL_MspInit (  
    void )
```

File Name : [stm32l1xx_hal_msp.c](#) Description : This file provides code for the MSP Initialization and de-Initialization codes.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause Initializes the Global MSP.

4.13 stm32l1xx_it.c File Reference

Interrupt Service Routines.

```
#include "main.h"
#include "stm32l1xx_it.h"
```

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 [SVC_Handler](#) (void)
This function handles System service call via SWI instruction.
- void [DebugMon_Handler](#) (void)
This function handles Debug monitor.
- void [PendSV_Handler](#) (void)
This function handles Pendable request for system service.
- void [SysTick_Handler](#) (void)
This function handles System tick timer.
- void [ADC1_IRQHandler](#) (void)
This function handles ADC global interrupt.
- void [EXTI9_5_IRQHandler](#) (void)
This function handles EXTI line[9:5] interrupts.
- void [USART1_IRQHandler](#) (void)
This function handles USART1 global interrupt.
- void [USART2_IRQHandler](#) (void)
This function handles USART2 global interrupt.
- void [USART3_IRQHandler](#) (void)
This function handles USART3 global interrupt.
- void [EXTI15_10_IRQHandler](#) (void)
This function handles EXTI line[15:10] interrupts.

Variables

- ADC_HandleTypeDef [hadc](#)
- UART_HandleTypeDef [huart1](#)
- UART_HandleTypeDef [huart2](#)
- UART_HandleTypeDef [huart3](#)

4.13.1 Detailed Description

Interrupt Service Routines.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.13.2 Function Documentation

4.13.2.1 ADC1_IRQHandler()

```
void ADC1_IRQHandler (  
    void )
```

This function handles ADC global interrupt.

4.13.2.2 BusFault_Handler()

```
void BusFault_Handler (  
    void )
```

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

4.13.2.3 DebugMon_Handler()

```
void DebugMon_Handler (
    void )
```

This function handles Debug monitor.

4.13.2.4 EXTI15_10_IRQHandler()

```
void EXTI15_10_IRQHandler (
    void )
```

This function handles EXTI line[15:10] interrupts.

4.13.2.5 EXTI9_5_IRQHandler()

```
void EXTI9_5_IRQHandler (
    void )
```

This function handles EXTI line[9:5] interrupts.

4.13.2.6 HardFault_Handler()

```
void HardFault_Handler (
    void )
```

This function handles Hard fault interrupt.

4.13.2.7 MemManage_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

4.13.2.8 NMI_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

4.13.2.9 PendSV_Handler()

```
void PendSV_Handler (  
    void )
```

This function handles Pendable request for system service.

4.13.2.10 SVC_Handler()

```
void SVC_Handler (  
    void )
```

This function handles System service call via SWI instruction.

4.13.2.11 SysTick_Handler()

```
void SysTick_Handler (  
    void )
```

This function handles System tick timer.

4.13.2.12 UsageFault_Handler()

```
void UsageFault_Handler (  
    void )
```

This function handles Undefined instruction or illegal state.

4.13.2.13 USART1_IRQHandler()

```
void USART1_IRQHandler (  
    void )
```

This function handles USART1 global interrupt.

4.13.2.14 USART2_IRQHandler()

```
void USART2_IRQHandler (  
    void )
```

This function handles USART2 global interrupt.

4.13.2.15 USART3_IRQHandler()

```
void USART3_IRQHandler (
    void )
```

This function handles USART3 global interrupt.

4.13.3 Variable Documentation

4.13.3.1 hadc

```
ADC_HandleTypeDef hadc
```

File Name : [ADC.c](#) Description : This file provides code for the configuration of the ADC instances.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.13.3.2 huart1

```
UART_HandleTypeDef huart1
```

File Name : [USART.c](#) Description : This file provides code for the configuration of the USART instances.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.13.3.3 huart2

```
UART_HandleTypeDef huart2
```

4.13.3.4 huart3

```
UART_HandleTypeDef huart3
```

4.14 stm32l1xx_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 [SVC_Handler](#) (void)
This function handles System service call via SWI instruction.
- void [DebugMon_Handler](#) (void)
This function handles Debug monitor.
- void [PendSV_Handler](#) (void)
This function handles Pendable request for system service.
- void [SysTick_Handler](#) (void)
This function handles System tick timer.
- void [ADC1_IRQHandler](#) (void)
This function handles ADC global interrupt.
- void [EXTI9_5_IRQHandler](#) (void)
This function handles EXTI line[9:5] interrupts.
- void [USART1_IRQHandler](#) (void)
This function handles USART1 global interrupt.
- void [USART2_IRQHandler](#) (void)
This function handles USART2 global interrupt.
- void [USART3_IRQHandler](#) (void)
This function handles USART3 global interrupt.
- void [EXTI15_10_IRQHandler](#) (void)
This function handles EXTI line[15:10] interrupts.

4.14.1 Detailed Description

This file contains the headers of the interrupt handlers.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.14.2 Function Documentation

4.14.2.1 ADC1_IRQHandler()

```
void ADC1_IRQHandler (
    void )
```

This function handles ADC global interrupt.

4.14.2.2 BusFault_Handler()

```
void BusFault_Handler (
    void )
```

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

4.14.2.3 DebugMon_Handler()

```
void DebugMon_Handler (
    void )
```

This function handles Debug monitor.

4.14.2.4 EXT_I15_10_IRQHandler()

```
void EXT_I15_10_IRQHandler (
    void )
```

This function handles EXTI line[15:10] interrupts.

4.14.2.5 EXT_I9_5_IRQHandler()

```
void EXT_I9_5_IRQHandler (
    void )
```

This function handles EXTI line[9:5] interrupts.

4.14.2.6 HardFault_Handler()

```
void HardFault_Handler (
    void )
```

This function handles Hard fault interrupt.

4.14.2.7 MemManage_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

4.14.2.8 NMI_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

4.14.2.9 PendSV_Handler()

```
void PendSV_Handler (
    void )
```

This function handles Pendable request for system service.

4.14.2.10 SVC_Handler()

```
void SVC_Handler (  
    void )
```

This function handles System service call via SWI instruction.

4.14.2.11 SysTick_Handler()

```
void SysTick_Handler (  
    void )
```

This function handles System tick timer.

4.14.2.12 UsageFault_Handler()

```
void UsageFault_Handler (  
    void )
```

This function handles Undefined instruction or illegal state.

4.14.2.13 USART1_IRQHandler()

```
void USART1_IRQHandler (  
    void )
```

This function handles USART1 global interrupt.

4.14.2.14 USART2_IRQHandler()

```
void USART2_IRQHandler (  
    void )
```

This function handles USART2 global interrupt.

4.14.2.15 USART3_IRQHandler()

```
void USART3_IRQHandler (  
    void )
```

This function handles USART3 global interrupt.

4.15 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

- int [errno](#)
- char ** [environ](#) = [__env](#)

4.15.1 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 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.15.2 Function Documentation

4.15.2.1 __attribute__()

```
__attribute__ (  
    (weak) )
```

4.15.2.2 __io_getchar()

```
int __io_getchar (  
    void )
```

4.15.2.3 __io_putchar()

```
int __io_putchar (  
    int ch )
```

4.15.2.4 _close()

```
int _close (  
    int file )
```

4.15.2.5 _execve()

```
int _execve (  
    char * name,  
    char ** argv,  
    char ** env )
```

4.15.2.6 `_exit()`

```
void _exit (  
    int status )
```

4.15.2.7 `_fork()`

```
int _fork (  
    void )
```

4.15.2.8 `_fstat()`

```
int _fstat (  
    int file,  
    struct stat * st )
```

4.15.2.9 `_getpid()`

```
int _getpid (  
    void )
```

4.15.2.10 `_isatty()`

```
int _isatty (  
    int file )
```

4.15.2.11 `_kill()`

```
int _kill (  
    int pid,  
    int sig )
```

4.15.2.12 `_link()`

```
int _link (
    char * old,
    char * new )
```

4.15.2.13 `_lseek()`

```
int _lseek (
    int file,
    int ptr,
    int dir )
```

4.15.2.14 `_open()`

```
int _open (
    char * path,
    int flags,
    ... )
```

4.15.2.15 `_stat()`

```
int _stat (
    char * file,
    struct stat * st )
```

4.15.2.16 `_times()`

```
int _times (
    struct tms * buf )
```

4.15.2.17 `_unlink()`

```
int _unlink (
    char * name )
```

4.15.2.18 `_wait()`

```
int _wait (
    int * status )
```

4.15.2.19 `initialise_monitor_handles()`

```
void initialise_monitor_handles ( )
```

4.15.3 Variable Documentation

4.15.3.1 `environ`

```
char** environ = __env
```

4.15.3.2 `errno`

```
int errno
```

4.16 `sysmem.c` File Reference

STM32CubeIDE Minimal System Memory calls file.

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

Functions

- register char *stack_ptr [asm](#) ("sp")
- caddr_t [_sbrk](#) (int incr)

Variables

- int [errno](#)

4.16.1 Detailed Description

STM32CubeIDE Minimal System Memory 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 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.16.2 Function Documentation

4.16.2.1 `_sbrk()`

```
caddr_t _sbrk (
    int incr )
```

`_sbrk` Increase program data space. Malloc and related functions depend on this

4.16.2.2 `asm()`

```
register char* stack_ptr asm (
    "sp" )
```

4.16.3 Variable Documentation

4.16.3.1 `errno`

```
int errno
```

4.17 system_stm32l1xx.c File Reference

CMSIS Cortex-M3 Device Peripheral Access Layer System Source File.

```
#include "stm32l1xx.h"
```

Macros

- #define [HSE_VALUE](#) ((uint32_t)8000000U)
- #define [HSI_VALUE](#) ((uint32_t)8000000U)
- #define [VECT_TAB_OFFSET](#) 0x00U

Functions

- void [SystemInit](#) (void)
Setup the microcontroller system. Initialize the Embedded Flash Interface, the PLL and update the SystemCoreClock variable.
- void [SystemCoreClockUpdate](#) (void)
Update SystemCoreClock 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](#) = 2097000U
- const uint8_t [PLLMulTable](#) [9] = {3U, 4U, 6U, 8U, 12U, 16U, 24U, 32U, 48U}
- const uint8_t [AHBPrescTable](#) [16] = {0U, 0U, 0U, 0U, 0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U, 6U, 7U, 8U, 9U}
- const uint8_t [APBPrescTable](#) [8] = {0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U}

4.17.1 Detailed Description

CMSIS Cortex-M3 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_stm32l1xx.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 component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.18 usart.c File Reference

```
#include "usart.h"
```

Functions

- void [MX_USART1_UART_Init](#) (void)
- void [MX_USART2_UART_Init](#) (void)
- void [MX_USART3_UART_Init](#) (void)
- void [HAL_UART_MspInit](#) (UART_HandleTypeDef *uartHandle)
- void [HAL_UART_MspDeInit](#) (UART_HandleTypeDef *uartHandle)

Variables

- UART_HandleTypeDef [huart1](#)
- UART_HandleTypeDef [huart2](#)
- UART_HandleTypeDef [huart3](#)

4.18.1 Function Documentation

4.18.1.1 HAL_UART_MspDeInit()

```
void HAL_UART_MspDeInit (  
    UART_HandleTypeDef * uartHandle )
```

USART1 GPIO Configuration
PA9 ----> USART1_TX PA10 ----> USART1_RX

USART2 GPIO Configuration
PA2 ----> USART2_TX PA3 ----> USART2_RX

USART3 GPIO Configuration
PB10 ----> USART3_TX PB11 ----> USART3_RX

4.18.1.2 HAL_UART_MspInit()

```
void HAL_UART_MspInit (
    UART_HandleTypeDef * uartHandle )
```

USART1 GPIO Configuration

PA9 ----> USART1_TX PA10 ----> USART1_RX

USART2 GPIO Configuration

PA2 ----> USART2_TX PA3 ----> USART2_RX

USART3 GPIO Configuration

PB10 ----> USART3_TX PB11 ----> USART3_RX

4.18.1.3 MX_USART1_UART_Init()

```
void MX_USART1_UART_Init (
    void )
```

4.18.1.4 MX_USART2_UART_Init()

```
void MX_USART2_UART_Init (
    void )
```

4.18.1.5 MX_USART3_UART_Init()

```
void MX_USART3_UART_Init (
    void )
```

4.18.2 Variable Documentation

4.18.2.1 huart1

```
UART_HandleTypeDef huart1
```

File Name : [USART.c](#) Description : This file provides code for the configuration of the USART instances.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.18.2.2 huart2

```
UART_HandleTypeDef huart2
```

4.18.2.3 huart3

```
UART_HandleTypeDef huart3
```

4.19 usart.h File Reference

```
#include "main.h"
```

Functions

- void [MX_USART1_UART_Init](#) (void)
- void [MX_USART2_UART_Init](#) (void)
- void [MX_USART3_UART_Init](#) (void)

Variables

- UART_HandleTypeDef [huart1](#)
- UART_HandleTypeDef [huart2](#)
- UART_HandleTypeDef [huart3](#)

4.19.1 Function Documentation

4.19.1.1 MX_USART1_UART_Init()

```
void MX_USART1_UART_Init (  
    void )
```

4.19.1.2 MX_USART2_UART_Init()

```
void MX_USART2_UART_Init (  
    void )
```

4.19.1.3 MX_USART3_UART_Init()

```
void MX_USART3_UART_Init (  
    void )
```

4.19.2 Variable Documentation

4.19.2.1 huart1

```
UART_HandleTypeDef huart1
```

File Name : [USART.h](#) Description : This file provides code for the configuration of the USART instances.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

File Name : [USART.c](#) Description : This file provides code for the configuration of the USART instances.

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

4.19.2.2 huart2

```
UART_HandleTypeDef huart2
```

4.19.2.3 huart3

```
UART_HandleTypeDef huart3
```

Index

- `__attribute__`
 - `syscalls.c`, [51](#)
 - `__io_getchar`
 - `syscalls.c`, [51](#)
 - `__io_putchar`
 - `syscalls.c`, [51](#)
 - `_close`
 - `syscalls.c`, [51](#)
 - `_execve`
 - `syscalls.c`, [51](#)
 - `_exit`
 - `syscalls.c`, [51](#)
 - `_fork`
 - `syscalls.c`, [52](#)
 - `_fstat`
 - `syscalls.c`, [52](#)
 - `_getpid`
 - `syscalls.c`, [52](#)
 - `_isatty`
 - `syscalls.c`, [52](#)
 - `_kill`
 - `syscalls.c`, [52](#)
 - `_link`
 - `syscalls.c`, [52](#)
 - `_lseek`
 - `syscalls.c`, [53](#)
 - `_open`
 - `syscalls.c`, [53](#)
 - `_sbrk`
 - `sysmem.c`, [55](#)
 - `_stat`
 - `syscalls.c`, [53](#)
 - `_times`
 - `syscalls.c`, [53](#)
 - `_unlink`
 - `syscalls.c`, [53](#)
 - `_wait`
 - `syscalls.c`, [53](#)
- `able`
 - `main.h`, [28](#)
- `adc.c`, [15](#)
 - `hadc`, [16](#)
 - `HAL_ADC_MspDeInit`, [15](#)
 - `HAL_ADC_MspInit`, [15](#)
 - `MX_ADC_Init`, [15](#)
- `adc.h`, [16](#)
 - `hadc`, [17](#)
 - `MX_ADC_Init`, [17](#)
- `ADC1_IRQHandler`
 - `stm32l1xx_it.c`, [42](#)
 - `stm32l1xx_it.h`, [47](#)
- `admin`
 - `motor.h`, [31](#)
- `admin_pc`
 - `motor.c`, [29](#)
 - `motor.h`, [30](#)
- `AHBPrescTable`
 - `STM32L1xx_System_Private_Variables`, [11](#)
- `APBPrescTable`
 - `STM32L1xx_System_Private_Variables`, [11](#)
- `asm`
 - `sysmem.c`, [55](#)
- `assert_param`
 - `stm32l1xx_hal_conf.h`, [34](#)
- `auto_down`
 - `motor.h`, [31](#)
- `auto_fotodetector`
 - `motor.c`, [29](#)
 - `motor.h`, [30](#)
- `auto_up`
 - `motor.h`, [31](#)
- `B1_EXTI_IRQn`
 - `main.h`, [24](#)
- `B1_GPIO_Port`
 - `main.h`, [24](#)
- `B1_Pin`
 - `main.h`, [24](#)
- `BusFault_Handler`
 - `stm32l1xx_it.c`, [42](#)
 - `stm32l1xx_it.h`, [47](#)
- `CMSIS`, [5](#)
- `control`
 - `motor.h`, [32](#)
- `count_position`
 - `motor.h`, [32](#)
- `data`
 - `main.h`, [28](#)
- `DATA_CACHE_ENABLE`
 - `stm32l1xx_hal_conf.h`, [34](#)
- `DebugMon_Handler`
 - `stm32l1xx_it.c`, [42](#)
 - `stm32l1xx_it.h`, [47](#)
- `dir`
 - `motor.h`, [32](#)
- `DIR_GPIO_Port`
 - `main.h`, [24](#)

DIR_Pin
 main.h, [24](#)
 ENABLE_GPIO_Port
 main.h, [25](#)
 ENABLE_Pin
 main.h, [25](#)
 END_DOWN_EXTI_IRQn
 main.h, [25](#)
 END_DOWN_GPIO_Port
 main.h, [25](#)
 END_DOWN_Pin
 main.h, [25](#)
 END_HIGH_EXTI_IRQn
 main.h, [25](#)
 END_HIGH_GPIO_Port
 main.h, [25](#)
 END_HIGH_Pin
 main.h, [25](#)
 environ
 syscalls.c, [54](#)
 errno
 syscalls.c, [54](#)
 sysmem.c, [55](#)
 Error_Handler
 main.c, [21](#)
 main.h, [28](#)
 EXTI15_10_IRQHandler
 stm32l1xx_it.c, [43](#)
 stm32l1xx_it.h, [47](#)
 EXTI9_5_IRQHandler
 stm32l1xx_it.c, [43](#)
 stm32l1xx_it.h, [48](#)

 fotodetector.c, [18](#)
 get_value_fotodetector, [18](#)
 fotodetector.h, [18](#)
 fotodetector_value, [19](#)
 get_value_fotodetector, [18](#)
 FOTODETECTOR_GPIO_Port
 main.h, [26](#)
 FOTODETECTOR_Pin
 main.h, [26](#)
 fotodetector_value
 fotodetector.h, [19](#)

 get_value_fotodetector
 fotodetector.c, [18](#)
 fotodetector.h, [18](#)
 gpio.c, [19](#)
 MX_GPIO_Init, [19](#)
 gpio.h, [20](#)
 MX_GPIO_Init, [20](#)

 hadc
 adc.c, [16](#)
 adc.h, [17](#)
 stm32l1xx_it.c, [45](#)
 HAL_ADC_MODULE_ENABLED
 stm32l1xx_hal_conf.h, [35](#)
 HAL_ADC_MspDeInit
 adc.c, [15](#)
 HAL_ADC_Msplnit
 adc.c, [15](#)
 HAL_CORTEX_MODULE_ENABLED
 stm32l1xx_hal_conf.h, [35](#)
 HAL_DMA_MODULE_ENABLED
 stm32l1xx_hal_conf.h, [35](#)
 HAL_FLASH_MODULE_ENABLED
 stm32l1xx_hal_conf.h, [35](#)
 HAL_GPIO_EXTI_Callback
 motor.c, [29](#)
 HAL_GPIO_MODULE_ENABLED
 stm32l1xx_hal_conf.h, [35](#)
 HAL_MODULE_ENABLED
 stm32l1xx_hal_conf.h, [35](#)
 HAL_Msplnit
 stm32l1xx_hal_msp.c, [40](#)
 HAL_PWR_MODULE_ENABLED
 stm32l1xx_hal_conf.h, [35](#)
 HAL_RCC_MODULE_ENABLED
 stm32l1xx_hal_conf.h, [36](#)
 HAL_UART_MODULE_ENABLED
 stm32l1xx_hal_conf.h, [36](#)
 HAL_UART_MspDeInit
 usart.c, [57](#)
 HAL_UART_Msplnit
 usart.c, [57](#)
 HAL_UART_RxCpltCallback
 main.c, [22](#)
 HardFault_Handler
 stm32l1xx_it.c, [43](#)
 stm32l1xx_it.h, [48](#)
 home_down
 motor.c, [29](#)
 motor.h, [31](#)
 home_up
 motor.c, [29](#)
 motor.h, [31](#)
 HSE_STARTUP_TIMEOUT
 stm32l1xx_hal_conf.h, [36](#)
 HSE_VALUE
 stm32l1xx_hal_conf.h, [36](#)
 STM32L1xx_System_Private_Defines, [9](#)
 HSI_VALUE
 stm32l1xx_hal_conf.h, [36](#)
 STM32L1xx_System_Private_Defines, [9](#)
 huart1
 stm32l1xx_it.c, [45](#)
 usart.c, [58](#)
 usart.h, [60](#)
 huart2
 stm32l1xx_it.c, [45](#)
 usart.c, [59](#)
 usart.h, [60](#)
 huart3
 stm32l1xx_it.c, [46](#)

- usart.c, [59](#)
- usart.h, [60](#)
- initialise_monitor_handles
 - syscalls.c, [54](#)
- INSTRUCTION_CACHE_ENABLE
 - stm32l1xx_hal_conf.h, [36](#)
- LD2_GPIO_Port
 - main.h, [26](#)
- LD2_Pin
 - main.h, [26](#)
- LSE_STARTUP_TIMEOUT
 - stm32l1xx_hal_conf.h, [36](#)
- LSE_VALUE
 - stm32l1xx_hal_conf.h, [37](#)
- LSI_VALUE
 - stm32l1xx_hal_conf.h, [37](#)
- main
 - main.c, [22](#)
- main.c, [21](#)
 - Error_Handler, [21](#)
 - HAL_UART_RxCpltCallback, [22](#)
 - main, [22](#)
 - SystemClock_Config, [22](#)
- main.h, [23](#)
 - able, [28](#)
 - B1_EXTI_IRQn, [24](#)
 - B1_GPIO_Port, [24](#)
 - B1_Pin, [24](#)
 - data, [28](#)
 - DIR_GPIO_Port, [24](#)
 - DIR_Pin, [24](#)
 - ENABLE_GPIO_Port, [25](#)
 - ENABLE_Pin, [25](#)
 - END_DOWN_EXTI_IRQn, [25](#)
 - END_DOWN_GPIO_Port, [25](#)
 - END_DOWN_Pin, [25](#)
 - END_HIGH_EXTI_IRQn, [25](#)
 - END_HIGH_GPIO_Port, [25](#)
 - END_HIGH_Pin, [25](#)
 - Error_Handler, [28](#)
 - FOTODETECTOR_GPIO_Port, [26](#)
 - FOTODETECTOR_Pin, [26](#)
 - LD2_GPIO_Port, [26](#)
 - LD2_Pin, [26](#)
 - STEP_GPIO_Port, [26](#)
 - STEP_Pin, [26](#)
 - SWO_GPIO_Port, [26](#)
 - SWO_Pin, [26](#)
 - TCK_GPIO_Port, [27](#)
 - TCK_Pin, [27](#)
 - TMS_GPIO_Port, [27](#)
 - TMS_Pin, [27](#)
 - USART_RX_GPIO_Port, [27](#)
 - USART_RX_Pin, [27](#)
 - USART_TX_GPIO_Port, [27](#)
 - USART_TX_Pin, [27](#)
- MemManage_Handler
 - stm32l1xx_it.c, [43](#)
 - stm32l1xx_it.h, [48](#)
- motor.c, [28](#)
 - admin_pc, [29](#)
 - auto_fotodetector, [29](#)
 - HAL_GPIO_EXTI_Callback, [29](#)
 - home_down, [29](#)
 - home_up, [29](#)
 - move_motor, [29](#)
 - new_data, [30](#)
- motor.h, [30](#)
 - admin, [31](#)
 - admin_pc, [30](#)
 - auto_down, [31](#)
 - auto_fotodetector, [30](#)
 - auto_up, [31](#)
 - control, [32](#)
 - count_position, [32](#)
 - dir, [32](#)
 - home_down, [31](#)
 - home_up, [31](#)
 - move_motor, [31](#)
 - new_data, [31](#)
 - position, [32](#)
 - set_position, [32](#)
- move_motor
 - motor.c, [29](#)
 - motor.h, [31](#)
- MSI_VALUE
 - stm32l1xx_hal_conf.h, [37](#)
- MX_ADC_Init
 - adc.c, [15](#)
 - adc.h, [17](#)
- MX_GPIO_Init
 - gpio.c, [19](#)
 - gpio.h, [20](#)
- MX_USART1_UART_Init
 - usart.c, [58](#)
 - usart.h, [59](#)
- MX_USART2_UART_Init
 - usart.c, [58](#)
 - usart.h, [59](#)
- MX_USART3_UART_Init
 - usart.c, [58](#)
 - usart.h, [60](#)
- new_data
 - motor.c, [30](#)
 - motor.h, [31](#)
- NMI_Handler
 - stm32l1xx_it.c, [43](#)
 - stm32l1xx_it.h, [48](#)
- PendSV_Handler
 - stm32l1xx_it.c, [43](#)
 - stm32l1xx_it.h, [48](#)
- PLLMulTable
 - STM32L1xx_System_Private_Variables, [11](#)

- position
 - motor.h, [32](#)
- PREFETCH_ENABLE
 - stm32l1xx_hal_conf.h, [37](#)
- set_position
 - motor.h, [32](#)
- STEP_GPIO_Port
 - main.h, [26](#)
- STEP_Pin
 - main.h, [26](#)
- stm32l1xx_hal_conf.h, [32](#)
 - assert_param, [34](#)
 - DATA_CACHE_ENABLE, [34](#)
 - HAL_ADC_MODULE_ENABLED, [35](#)
 - HAL_CORTEX_MODULE_ENABLED, [35](#)
 - HAL_DMA_MODULE_ENABLED, [35](#)
 - HAL_FLASH_MODULE_ENABLED, [35](#)
 - HAL_GPIO_MODULE_ENABLED, [35](#)
 - HAL_MODULE_ENABLED, [35](#)
 - HAL_PWR_MODULE_ENABLED, [35](#)
 - HAL_RCC_MODULE_ENABLED, [36](#)
 - HAL_UART_MODULE_ENABLED, [36](#)
 - HSE_STARTUP_TIMEOUT, [36](#)
 - HSE_VALUE, [36](#)
 - HSI_VALUE, [36](#)
 - INSTRUCTION_CACHE_ENABLE, [36](#)
 - LSE_STARTUP_TIMEOUT, [36](#)
 - LSE_VALUE, [37](#)
 - LSI_VALUE, [37](#)
 - MSI_VALUE, [37](#)
 - PREFETCH_ENABLE, [37](#)
 - TICK_INT_PRIORITY, [37](#)
 - USE_HAL_ADC_REGISTER_CALLBACKS, [37](#)
 - USE_HAL_COMP_REGISTER_CALLBACKS, [38](#)
 - USE_HAL_DAC_REGISTER_CALLBACKS, [38](#)
 - USE_HAL_I2C_REGISTER_CALLBACKS, [38](#)
 - USE_HAL_I2S_REGISTER_CALLBACKS, [38](#)
 - USE_HAL_IRDA_REGISTER_CALLBACKS, [38](#)
 - USE_HAL_OPAMP_REGISTER_CALLBACKS, [38](#)
 - USE_HAL_PCD_REGISTER_CALLBACKS, [38](#)
 - USE_HAL_RTC_REGISTER_CALLBACKS, [39](#)
 - USE_HAL_SDMMC_REGISTER_CALLBACKS, [39](#)
 - USE_HAL_SMARTCARD_REGISTER_CALLBACKS, [39](#)
 - USE_HAL_SPI_REGISTER_CALLBACKS, [39](#)
 - USE_HAL_TIM_REGISTER_CALLBACKS, [39](#)
 - USE_HAL_UART_REGISTER_CALLBACKS, [39](#)
 - USE_HAL_USART_REGISTER_CALLBACKS, [39](#)
 - USE_HAL_WWDG_REGISTER_CALLBACKS, [39](#)
 - USE_RTOS, [40](#)
 - USE_SPI_CRC, [40](#)
 - VDD_VALUE, [40](#)
- stm32l1xx_hal_msp.c, [40](#)
 - HAL_MspInit, [40](#)
- stm32l1xx_it.c, [41](#)
 - ADC1_IRQHandler, [42](#)
 - BusFault_Handler, [42](#)
 - DebugMon_Handler, [42](#)
 - EXTI15_10_IRQHandler, [43](#)
 - EXTI9_5_IRQHandler, [43](#)
 - hadc, [45](#)
 - HardFault_Handler, [43](#)
 - huart1, [45](#)
 - huart2, [45](#)
 - huart3, [46](#)
 - MemManage_Handler, [43](#)
 - NMI_Handler, [43](#)
 - PendSV_Handler, [43](#)
 - SVC_Handler, [44](#)
 - SysTick_Handler, [44](#)
 - UsageFault_Handler, [44](#)
 - USART1_IRQHandler, [44](#)
 - USART2_IRQHandler, [44](#)
 - USART3_IRQHandler, [44](#)
- stm32l1xx_it.h, [46](#)
 - ADC1_IRQHandler, [47](#)
 - BusFault_Handler, [47](#)
 - DebugMon_Handler, [47](#)
 - EXTI15_10_IRQHandler, [47](#)
 - EXTI9_5_IRQHandler, [48](#)
 - HardFault_Handler, [48](#)
 - MemManage_Handler, [48](#)
 - NMI_Handler, [48](#)
 - PendSV_Handler, [48](#)
 - SVC_Handler, [48](#)
 - SysTick_Handler, [49](#)
 - UsageFault_Handler, [49](#)
 - USART1_IRQHandler, [49](#)
 - USART2_IRQHandler, [49](#)
 - USART3_IRQHandler, [49](#)
- Stm32l1xx_system, [6](#)
- STM32L1xx_System_Private_Defines, [9](#)
 - HSE_VALUE, [9](#)
 - HSI_VALUE, [9](#)
 - VECT_TAB_OFFSET, [9](#)
- STM32L1xx_System_Private_FunctionPrototypes, [12](#)
- STM32L1xx_System_Private_Functions, [13](#)
 - SystemCoreClockUpdate, [13](#)
 - SystemInit, [14](#)
- STM32L1xx_System_Private_Includes, [7](#)
- STM32L1xx_System_Private_Macros, [10](#)
- STM32L1xx_System_Private_TypesDefinitions, [8](#)
- STM32L1xx_System_Private_Variables, [11](#)
 - AHBPrescTable, [11](#)
 - APBPrescTable, [11](#)
 - PLLMulTable, [11](#)
 - SystemCoreClock, [11](#)
- SVC_Handler
 - stm32l1xx_it.c, [44](#)
 - stm32l1xx_it.h, [48](#)
- SWO_GPIO_Port
 - main.h, [26](#)
- SWO_Pin
 - main.h, [26](#)
- syscalls.c, [50](#)

- [__attribute__, 51](#)
- [__io_getchar, 51](#)
- [__io_putchar, 51](#)
- [_close, 51](#)
- [_execve, 51](#)
- [_exit, 51](#)
- [_fork, 52](#)
- [_fstat, 52](#)
- [_getpid, 52](#)
- [_isatty, 52](#)
- [_kill, 52](#)
- [_link, 52](#)
- [_lseek, 53](#)
- [_open, 53](#)
- [_stat, 53](#)
- [_times, 53](#)
- [_unlink, 53](#)
- [_wait, 53](#)
- [environ, 54](#)
- [errno, 54](#)
- [initialise_monitor_handles, 54](#)
- [sysmem.c, 54](#)
 - [_sbrk, 55](#)
 - [asm, 55](#)
 - [errno, 55](#)
- [system_stm32l1xx.c, 56](#)
- [SystemClock_Config](#)
 - [main.c, 22](#)
- [SystemCoreClock](#)
 - [STM32L1xx_System_Private_Variables, 11](#)
- [SystemCoreClockUpdate](#)
 - [STM32L1xx_System_Private_Functions, 13](#)
- [SystemInit](#)
 - [STM32L1xx_System_Private_Functions, 14](#)
- [SysTick_Handler](#)
 - [stm32l1xx_it.c, 44](#)
 - [stm32l1xx_it.h, 49](#)
- [TCK_GPIO_Port](#)
 - [main.h, 27](#)
- [TCK_Pin](#)
 - [main.h, 27](#)
- [TICK_INT_PRIORITY](#)
 - [stm32l1xx_hal_conf.h, 37](#)
- [TMS_GPIO_Port](#)
 - [main.h, 27](#)
- [TMS_Pin](#)
 - [main.h, 27](#)
- [UsageFault_Handler](#)
 - [stm32l1xx_it.c, 44](#)
 - [stm32l1xx_it.h, 49](#)
- [usart.c, 57](#)
 - [HAL_UART_MspDeInit, 57](#)
 - [HAL_UART_MspInit, 57](#)
 - [huart1, 58](#)
 - [huart2, 59](#)
 - [huart3, 59](#)
 - [MX_USART1_UART_Init, 58](#)
 - [MX_USART2_UART_Init, 58](#)
 - [MX_USART3_UART_Init, 58](#)
- [usart.h, 59](#)
 - [huart1, 60](#)
 - [huart2, 60](#)
 - [huart3, 60](#)
 - [MX_USART1_UART_Init, 59](#)
 - [MX_USART2_UART_Init, 59](#)
 - [MX_USART3_UART_Init, 60](#)
- [USART1_IRQHandler](#)
 - [stm32l1xx_it.c, 44](#)
 - [stm32l1xx_it.h, 49](#)
- [USART2_IRQHandler](#)
 - [stm32l1xx_it.c, 44](#)
 - [stm32l1xx_it.h, 49](#)
- [USART3_IRQHandler](#)
 - [stm32l1xx_it.c, 44](#)
 - [stm32l1xx_it.h, 49](#)
- [USART_RX_GPIO_Port](#)
 - [main.h, 27](#)
- [USART_RX_Pin](#)
 - [main.h, 27](#)
- [USART_TX_GPIO_Port](#)
 - [main.h, 27](#)
- [USART_TX_Pin](#)
 - [main.h, 27](#)
- [USE_HAL_ADC_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 37](#)
- [USE_HAL_COMP_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 38](#)
- [USE_HAL_DAC_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 38](#)
- [USE_HAL_I2C_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 38](#)
- [USE_HAL_I2S_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 38](#)
- [USE_HAL_IRDA_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 38](#)
- [USE_HAL_OPAMP_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 38](#)
- [USE_HAL_PCD_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 38](#)
- [USE_HAL_RTC_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 39](#)
- [USE_HAL_SDMMC_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 39](#)
- [USE_HAL_SMARTCARD_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 39](#)
- [USE_HAL_SPI_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 39](#)
- [USE_HAL_TIM_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 39](#)
- [USE_HAL_UART_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 39](#)
- [USE_HAL_USART_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 39](#)
- [USE_HAL_WWDG_REGISTER_CALLBACKS](#)
 - [stm32l1xx_hal_conf.h, 39](#)

USE_RTOS

stm32l1xx_hal_conf.h, [40](#)

USE_SPI_CRC

stm32l1xx_hal_conf.h, [40](#)

VDD_VALUE

stm32l1xx_hal_conf.h, [40](#)

VECT_TAB_OFFSET

STM32L1xx_System_Private_Defines, [9](#)