

the Koopa Slingers

Generated by Doxygen 1.10.0



<b>1 MAIN PAGE</b>	<b>1</b>
1.1 Goals	1
1.2 Requirements	1
1.2.1 The Electronics Requirements:	1
1.2.2 The Manufacturing Requirements:	2
1.2.3 The Safety Requirements:	2
1.3 Hardware	2
1.4 Software	2
1.5 REPOSITORY REFERENCE	2
1.6 CONTACT INFO	3
<b>2 Hardware Documentation</b>	<b>5</b>
2.1 Introduction	5
2.2 Initial Concept	5
2.3 Bill of Materials	5
2.4 Schematic	5
2.4.0.1 Custom PCB Schematics	5
2.4.0.2 Controller Glove PCB	6
2.5 Common Challenges	6
<b>3 Software Documentation</b>	<b>9</b>
3.1 Introduction	9
3.2 Finite State Machine	9
3.3 Code	9
<b>4 Topic Index</b>	<b>11</b>
4.1 Topics	11
<b>5 Data Structure Index</b>	<b>13</b>
5.1 Data Structures	13
<b>6 File Index</b>	<b>15</b>
6.1 File List	15
<b>7 Topic Documentation</b>	<b>17</b>
7.1 CMSIS	17
7.1.1 Detailed Description	17
7.1.2 Stm32f4xx_system	17
7.1.2.1 Detailed Description	17
7.1.2.2 STM32F4xx_System_Private_Includes	17
7.1.2.3 STM32F4xx_System_Private_TypesDefinitions	19
7.1.2.4 STM32F4xx_System_Private_Defines	19
7.1.2.5 STM32F4xx_System_Private_Macros	19
7.1.2.6 STM32F4xx_System_Private_Variables	19

7.1.2.7 STM32F4xx_System_Private_FunctionPrototypes . . . . .	20
7.1.2.8 STM32F4xx_System_Private_Functions . . . . .	20
<b>8 Data Structure Documentation</b>	<b>23</b>
8.1 D4215X Struct Reference . . . . .	23
8.1.1 Detailed Description . . . . .	23
8.1.2 Field Documentation . . . . .	23
8.1.2.1 channel . . . . .	23
8.1.2.2 speed . . . . .	23
8.1.2.3 timer . . . . .	24
8.2 LEDX Struct Reference . . . . .	24
8.2.1 Detailed Description . . . . .	24
8.2.2 Field Documentation . . . . .	24
8.2.2.1 GPIOx . . . . .	24
8.2.2.2 PIN . . . . .	24
8.3 MPU6050 Struct Reference . . . . .	25
8.3.1 Detailed Description . . . . .	25
8.3.2 Field Documentation . . . . .	25
8.3.2.1 addr . . . . .	25
8.3.2.2 dt . . . . .	25
8.3.2.3 gX . . . . .	25
8.3.2.4 gX_offset . . . . .	25
8.3.2.5 gY . . . . .	26
8.3.2.6 gY_offset . . . . .	26
8.3.2.7 gZ . . . . .	26
8.3.2.8 gZ_offset . . . . .	26
8.3.2.9 hi2c . . . . .	26
8.3.2.10 status . . . . .	26
8.4 RadioX Struct Reference . . . . .	26
8.4.1 Detailed Description . . . . .	27
8.4.2 Field Documentation . . . . .	27
8.4.2.1 channel1 . . . . .	27
8.4.2.2 channel2 . . . . .	27
8.4.2.3 counter1 . . . . .	27
8.4.2.4 counter2 . . . . .	27
8.4.2.5 pulseWidth . . . . .	27
8.4.2.6 sum1 . . . . .	28
8.4.2.7 sum2 . . . . .	28
8.4.2.8 timer . . . . .	28
8.5 StepperX Struct Reference . . . . .	28
8.5.1 Detailed Description . . . . .	28
8.5.2 Field Documentation . . . . .	28

8.5.2.1 DIR_PIN . . . . .	28
8.5.2.2 EN_PIN . . . . .	29
8.5.2.3 GPIOx . . . . .	29
8.5.2.4 STP_PIN . . . . .	29
8.6 SwitchX Struct Reference . . . . .	29
8.6.1 Detailed Description . . . . .	29
8.6.2 Field Documentation . . . . .	29
8.6.2.1 GPIOx . . . . .	29
8.6.2.2 PIN . . . . .	30
8.6.2.3 status . . . . .	30
<b>9 File Documentation</b>	<b>31</b>
9.1 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/main_C.h File Reference . . . . .	31
9.1.1 Detailed Description . . . . .	31
9.1.2 Function Documentation . . . . .	31
9.1.2.1 Error_Handler() . . . . .	31
9.2 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/stm32f4xx_hal_conf_C.h File Reference	32
9.2.1 Detailed Description . . . . .	34
9.2.2 Macro Definition Documentation . . . . .	35
9.2.2.1 assert_param . . . . .	35
9.2.2.2 DATA_CACHE_ENABLE . . . . .	35
9.2.2.3 DP83848_PHY_ADDRESS . . . . .	35
9.2.2.4 ETH_RX_BUF_SIZE . . . . .	35
9.2.2.5 ETH_RXBUFNB . . . . .	35
9.2.2.6 ETH_TX_BUF_SIZE . . . . .	35
9.2.2.7 ETH_TXBUFNB . . . . .	35
9.2.2.8 EXTERNAL_CLOCK_VALUE . . . . .	35
9.2.2.9 HAL_CORTEX_MODULE_ENABLED . . . . .	36
9.2.2.10 HAL_DMA_MODULE_ENABLED . . . . .	36
9.2.2.11 HAL_EXTI_MODULE_ENABLED . . . . .	36
9.2.2.12 HAL_FLASH_MODULE_ENABLED . . . . .	36
9.2.2.13 HAL_GPIO_MODULE_ENABLED . . . . .	36
9.2.2.14 HAL_I2C_MODULE_ENABLED . . . . .	36
9.2.2.15 HAL_MODULE_ENABLED . . . . .	36
9.2.2.16 HAL_PWR_MODULE_ENABLED . . . . .	36
9.2.2.17 HAL_RCC_MODULE_ENABLED . . . . .	36
9.2.2.18 HAL_UART_MODULE_ENABLED . . . . .	36
9.2.2.19 HSE_STARTUP_TIMEOUT . . . . .	37
9.2.2.20 HSE_VALUE . . . . .	37
9.2.2.21 HSI_VALUE . . . . .	37
9.2.2.22 INSTRUCTION_CACHE_ENABLE . . . . .	37
9.2.2.23 LSE_STARTUP_TIMEOUT . . . . .	37

9.2.2.24 LSE_VALUE . . . . .	37
9.2.2.25 LSI_VALUE . . . . .	38
9.2.2.26 MAC_ADDR0 . . . . .	38
9.2.2.27 MAC_ADDR1 . . . . .	38
9.2.2.28 MAC_ADDR2 . . . . .	38
9.2.2.29 MAC_ADDR3 . . . . .	38
9.2.2.30 MAC_ADDR4 . . . . .	38
9.2.2.31 MAC_ADDR5 . . . . .	38
9.2.2.32 PHY_AUTONEGO_COMPLETE . . . . .	38
9.2.2.33 PHY_AUTONEGOTIATION . . . . .	39
9.2.2.34 PHY_BCR . . . . .	39
9.2.2.35 PHY_BSR . . . . .	39
9.2.2.36 PHY_CONFIG_DELAY . . . . .	39
9.2.2.37 PHY_DUPLEX_STATUS . . . . .	39
9.2.2.38 PHY_FULLDUPLEX_100M . . . . .	39
9.2.2.39 PHY_FULLDUPLEX_10M . . . . .	39
9.2.2.40 PHY_HALFDUPLEX_100M . . . . .	40
9.2.2.41 PHY_HALFDUPLEX_10M . . . . .	40
9.2.2.42 PHY_ISOLATE . . . . .	40
9.2.2.43 PHY_JABBER_DETECTION . . . . .	40
9.2.2.44 PHY_LINKED_STATUS . . . . .	40
9.2.2.45 PHY_LOOPBACK . . . . .	40
9.2.2.46 PHY_POWERDOWN . . . . .	40
9.2.2.47 PHY_READ_TO . . . . .	41
9.2.2.48 PHY_RESET . . . . .	41
9.2.2.49 PHY_RESET_DELAY . . . . .	41
9.2.2.50 PHY_RESTART_AUTONEGOTIATION . . . . .	41
9.2.2.51 PHY_SPEED_STATUS . . . . .	41
9.2.2.52 PHY_SR . . . . .	41
9.2.2.53 PHY_WRITE_TO . . . . .	41
9.2.2.54 PREFETCH_ENABLE . . . . .	41
9.2.2.55 TICK_INT_PRIORITY . . . . .	42
9.2.2.56 USE_HAL_ADC_REGISTER_CALLBACKS . . . . .	42
9.2.2.57 USE_HAL_CAN_REGISTER_CALLBACKS . . . . .	42
9.2.2.58 USE_HAL_CEC_REGISTER_CALLBACKS . . . . .	42
9.2.2.59 USE_HAL_Cryp_REGISTER_CALLBACKS . . . . .	42
9.2.2.60 USE_HAL_DAC_REGISTER_CALLBACKS . . . . .	42
9.2.2.61 USE_HAL_DCMI_REGISTER_CALLBACKS . . . . .	42
9.2.2.62 USE_HAL_DFSDM_REGISTER_CALLBACKS . . . . .	42
9.2.2.63 USE_HAL_DMA2D_REGISTER_CALLBACKS . . . . .	42
9.2.2.64 USE_HAL_DSI_REGISTER_CALLBACKS . . . . .	42
9.2.2.65 USE_HAL_ETH_REGISTER_CALLBACKS . . . . .	43

9.2.2.66 USE_HAL_FMPI2C_REGISTER_CALLBACKS . . . . .	43
9.2.2.67 USE_HAL_FMPSPMBUS_REGISTER_CALLBACKS . . . . .	43
9.2.2.68 USE_HAL_HASH_REGISTER_CALLBACKS . . . . .	43
9.2.2.69 USE_HAL_HCD_REGISTER_CALLBACKS . . . . .	43
9.2.2.70 USE_HAL_I2C_REGISTER_CALLBACKS . . . . .	43
9.2.2.71 USE_HAL_I2S_REGISTER_CALLBACKS . . . . .	43
9.2.2.72 USE_HAL_IRDA_REGISTER_CALLBACKS . . . . .	43
9.2.2.73 USE_HAL_LPTIM_REGISTER_CALLBACKS . . . . .	43
9.2.2.74 USE_HAL_LTDC_REGISTER_CALLBACKS . . . . .	43
9.2.2.75 USE_HAL_MMC_REGISTER_CALLBACKS . . . . .	44
9.2.2.76 USE_HAL_NAND_REGISTER_CALLBACKS . . . . .	44
9.2.2.77 USE_HAL_NOR_REGISTER_CALLBACKS . . . . .	44
9.2.2.78 USE_HAL_PCCARD_REGISTER_CALLBACKS . . . . .	44
9.2.2.79 USE_HAL_PCD_REGISTER_CALLBACKS . . . . .	44
9.2.2.80 USE_HAL_QSPI_REGISTER_CALLBACKS . . . . .	44
9.2.2.81 USE_HAL_RNG_REGISTER_CALLBACKS . . . . .	44
9.2.2.82 USE_HAL_RTC_REGISTER_CALLBACKS . . . . .	44
9.2.2.83 USE_HAL_SAI_REGISTER_CALLBACKS . . . . .	44
9.2.2.84 USE_HAL_SD_REGISTER_CALLBACKS . . . . .	44
9.2.2.85 USE_HAL_SDRAM_REGISTER_CALLBACKS . . . . .	45
9.2.2.86 USE_HAL_SMARTCARD_REGISTER_CALLBACKS . . . . .	45
9.2.2.87 USE_HAL_SMBUS_REGISTER_CALLBACKS . . . . .	45
9.2.2.88 USE_HAL_SPDIFRX_REGISTER_CALLBACKS . . . . .	45
9.2.2.89 USE_HAL_SPI_REGISTER_CALLBACKS . . . . .	45
9.2.2.90 USE_HAL_SRAM_REGISTER_CALLBACKS . . . . .	45
9.2.2.91 USE_HAL_TIM_REGISTER_CALLBACKS . . . . .	45
9.2.2.92 USE_HAL_UART_REGISTER_CALLBACKS . . . . .	45
9.2.2.93 USE_HAL_USART_REGISTER_CALLBACKS . . . . .	45
9.2.2.94 USE_HAL_WWDG_REGISTER_CALLBACKS . . . . .	45
9.2.2.95 USE_RTOS . . . . .	46
9.2.2.96 USE_SPI_CRC . . . . .	46
9.2.2.97 VDD_VALUE . . . . .	46
9.3 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/stm32f4xx_it_C.h File Reference . . . .	46
9.3.1 Detailed Description . . . . .	47
9.3.2 Function Documentation . . . . .	47
9.3.2.1 BusFault_Handler() . . . . .	47
9.3.2.2 DebugMon_Handler() . . . . .	47
9.3.2.3 HardFault_Handler() . . . . .	47
9.3.2.4 MemManage_Handler() . . . . .	47
9.3.2.5 NMI_Handler() . . . . .	47
9.3.2.6 PendSV_Handler() . . . . .	48
9.3.2.7 SVC_Handler() . . . . .	48

9.3.2.8 SysTick_Handler()	48
9.3.2.9 UsageFault_Handler()	48
9.4 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/main_C.c File Reference	48
9.4.1 Detailed Description	49
9.4.2 Function Documentation	49
9.4.2.1 Error_Handler()	49
9.4.2.2 main()	49
9.4.2.3 SystemClock_Config()	50
9.4.3 Variable Documentation	50
9.4.3.1 buffer	50
9.4.3.2 gX	50
9.4.3.3 gY	50
9.4.3.4 gZ	50
9.4.3.5 hi2c1	50
9.4.3.6 huart1	51
9.4.3.7 huart2	51
9.4.3.8 move	51
9.4.3.9 shot	51
9.4.3.10 stat	51
9.4.3.11 STATE	51
9.4.3.12 STATE_0_INIT	51
9.4.3.13 STATE_1_ERROR	51
9.4.3.14 STATE_2_IMU	51
9.4.3.15 STATE_3_BUTTON_LED	51
9.4.3.16 STATE_4_TRANSFER	52
9.5 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/stm32f4xx_hal_msp_C.c File Reference	52
9.5.1 Detailed Description	52
9.5.2 Function Documentation	52
9.5.2.1 HAL_I2C_MspDeInit()	52
9.5.2.2 HAL_I2C_MspInit()	53
9.5.2.3 HAL_MspInit()	53
9.5.2.4 HAL_UART_MspDeInit()	53
9.5.2.5 HAL_UART_MspInit()	54
9.6 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/stm32f4xx_it_C.c File Reference	54
9.6.1 Detailed Description	55
9.6.2 Function Documentation	55
9.6.2.1 BusFault_Handler()	55
9.6.2.2 DebugMon_Handler()	55
9.6.2.3 HardFault_Handler()	55
9.6.2.4 MemManage_Handler()	56
9.6.2.5 NMI_Handler()	56
9.6.2.6 PendSV_Handler()	56



9.6.2.7 SVC_Handler()	56
9.6.2.8 SysTick_Handler()	56
9.6.2.9 UsageFault_Handler()	56
9.7 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/syscalls_C.c File Reference	57
9.7.1 Detailed Description	57
9.7.2 Function Documentation	58
9.7.2.1 __attribute__()	58
9.7.2.2 __io_getchar()	58
9.7.2.3 __io_putchar()	58
9.7.2.4 _close()	58
9.7.2.5 _execve()	58
9.7.2.6 _exit()	58
9.7.2.7 _fork()	58
9.7.2.8 _fstat()	58
9.7.2.9 _getpid()	59
9.7.2.10 _isatty()	59
9.7.2.11 _kill()	59
9.7.2.12 _link()	59
9.7.2.13 _lseek()	59
9.7.2.14 _open()	59
9.7.2.15 _stat()	59
9.7.2.16 _times()	59
9.7.2.17 _unlink()	60
9.7.2.18 _wait()	60
9.7.2.19 initialise_monitor_handles()	60
9.7.3 Variable Documentation	60
9.7.3.1 environ	60
9.8 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/systemem_C.c File Reference	60
9.8.1 Detailed Description	60
9.8.2 Function Documentation	61
9.8.2.1 _sbrk()	61
9.9 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/system_stm32f4xx_C.c File Reference	61
9.9.1 Detailed Description	62
9.10 Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/d4215.h File Reference	62
9.10.1 Detailed Description	63
9.10.2 Function Documentation	63
9.10.2.1 D4215_init()	63
9.10.2.2 D4215_set()	63
9.11 Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/led.h File Reference	64
9.11.1 Detailed Description	64
9.11.2 Function Documentation	64
9.11.2.1 LED_init()	64

9.11.2.2 LED_off()	65
9.11.2.3 LED_on()	65
9.11.2.4 LED_toggle()	65
9.12 Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/radio_driver.h File Reference	65
9.12.1 Detailed Description	66
9.12.2 Function Documentation	66
9.12.2.1 capturePulseWidth()	66
9.12.2.2 Radio_getPulseWidth()	66
9.12.2.3 Radio_init()	67
9.12.2.4 update()	67
9.13 Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/stepper_driver.h File Reference	67
9.13.1 Detailed Description	68
9.13.2 Function Documentation	68
9.13.2.1 Delay_us()	68
9.13.2.2 Stepper_disable()	68
9.13.2.3 Stepper_enable()	69
9.13.2.4 Stepper_init()	69
9.13.2.5 Stepper_setspeed()	69
9.13.2.6 SysTick_Init()	69
9.14 Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/switch.h File Reference	70
9.14.1 Detailed Description	70
9.14.2 Function Documentation	70
9.14.2.1 Switch_getStatus()	70
9.14.2.2 Switch_init()	71
9.15 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/main.h File Reference	71
9.15.1 Detailed Description	71
9.15.2 Function Documentation	71
9.15.2.1 Error_Handler()	71
9.16 Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/main.h File Reference	72
9.16.1 Detailed Description	72
9.16.2 Function Documentation	72
9.16.2.1 Error_Handler()	72
9.16.2.2 HAL_TIM_MspPostInit()	73
9.17 Ball_Launcher_Controller/Core/Inc/main.h File Reference	73
9.17.1 Detailed Description	73
9.17.2 Function Documentation	73
9.17.2.1 Error_Handler()	73
9.18 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/mpu6050.h File Reference	74
9.18.1 Detailed Description	74
9.18.2 Function Documentation	74
9.18.2.1 mpu6050_calibrate()	74
9.18.2.2 mpu6050_get_gX()	75

9.18.2.3 mpu6050_get_gY()	75
9.18.2.4 mpu6050_get_gZ()	75
9.18.2.5 mpu6050_get_X()	76
9.18.2.6 mpu6050_get_Y()	76
9.18.2.7 mpu6050_get_Z()	76
9.18.2.8 mpu6050_init()	77
9.18.2.9 mpu6050_update()	77
9.19 Ball_Launcher_Controller/Core/Inc/mpu6050.h File Reference	77
9.19.1 Detailed Description	78
9.19.2 Function Documentation	78
9.19.2.1 mpu6050_calibrate()	78
9.19.2.2 mpu6050_get_gX()	78
9.19.2.3 mpu6050_get_gY()	79
9.19.2.4 mpu6050_get_gZ()	79
9.19.2.5 mpu6050_get_X()	79
9.19.2.6 mpu6050_get_Y()	80
9.19.2.7 mpu6050_get_Z()	80
9.19.2.8 mpu6050_init()	80
9.19.2.9 mpu6050_update()	81
9.20 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/stm32f4xx_hal_conf.h File Reference	81
9.20.1 Macro Definition Documentation	83
9.20.1.1 assert_param	83
9.20.1.2 DATA_CACHE_ENABLE	84
9.20.1.3 DP83848_PHY_ADDRESS	84
9.20.1.4 ETH_RX_BUF_SIZE	84
9.20.1.5 ETH_RXBUFNB	84
9.20.1.6 ETH_TX_BUF_SIZE	84
9.20.1.7 ETH_TXBUFNB	84
9.20.1.8 EXTERNAL_CLOCK_VALUE	84
9.20.1.9 HAL_CORTEX_MODULE_ENABLED	84
9.20.1.10 HAL_DMA_MODULE_ENABLED	84
9.20.1.11 HAL_EXTI_MODULE_ENABLED	85
9.20.1.12 HAL_FLASH_MODULE_ENABLED	85
9.20.1.13 HAL_GPIO_MODULE_ENABLED	85
9.20.1.14 HAL_I2C_MODULE_ENABLED	85
9.20.1.15 HAL_MODULE_ENABLED	85
9.20.1.16 HAL_PWR_MODULE_ENABLED	85
9.20.1.17 HAL_RCC_MODULE_ENABLED	85
9.20.1.18 HAL_UART_MODULE_ENABLED	85
9.20.1.19 HSE_STARTUP_TIMEOUT	85
9.20.1.20 HSE_VALUE	86
9.20.1.21 HSI_VALUE	86

9.20.1.22 INSTRUCTION_CACHE_ENABLE . . . . .	86
9.20.1.23 LSE_STARTUP_TIMEOUT . . . . .	86
9.20.1.24 LSE_VALUE . . . . .	86
9.20.1.25 LSI_VALUE . . . . .	86
9.20.1.26 MAC_ADDR0 . . . . .	87
9.20.1.27 MAC_ADDR1 . . . . .	87
9.20.1.28 MAC_ADDR2 . . . . .	87
9.20.1.29 MAC_ADDR3 . . . . .	87
9.20.1.30 MAC_ADDR4 . . . . .	87
9.20.1.31 MAC_ADDR5 . . . . .	87
9.20.1.32 PHY_AUTONEGO_COMPLETE . . . . .	87
9.20.1.33 PHY_AUTONEGOTIATION . . . . .	87
9.20.1.34 PHY_BCR . . . . .	87
9.20.1.35 PHY_BSR . . . . .	88
9.20.1.36 PHY_CONFIG_DELAY . . . . .	88
9.20.1.37 PHY_DUPLEX_STATUS . . . . .	88
9.20.1.38 PHY_FULLDUPLEX_100M . . . . .	88
9.20.1.39 PHY_FULLDUPLEX_10M . . . . .	88
9.20.1.40 PHY_HALFDUPLEX_100M . . . . .	88
9.20.1.41 PHY_HALFDUPLEX_10M . . . . .	88
9.20.1.42 PHY_ISOLATE . . . . .	89
9.20.1.43 PHY_JABBER_DETECTION . . . . .	89
9.20.1.44 PHY_LINKED_STATUS . . . . .	89
9.20.1.45 PHY_LOOPBACK . . . . .	89
9.20.1.46 PHY_POWERDOWN . . . . .	89
9.20.1.47 PHY_READ_TO . . . . .	89
9.20.1.48 PHY_RESET . . . . .	89
9.20.1.49 PHY_RESET_DELAY . . . . .	90
9.20.1.50 PHY_RESTART_AUTONEGOTIATION . . . . .	90
9.20.1.51 PHY_SPEED_STATUS . . . . .	90
9.20.1.52 PHY_SR . . . . .	90
9.20.1.53 PHY_WRITE_TO . . . . .	90
9.20.1.54 PREFETCH_ENABLE . . . . .	90
9.20.1.55 TICK_INT_PRIORITY . . . . .	90
9.20.1.56 USE_HAL_ADC_REGISTER_CALLBACKS . . . . .	90
9.20.1.57 USE_HAL_CAN_REGISTER_CALLBACKS . . . . .	91
9.20.1.58 USE_HAL_CEC_REGISTER_CALLBACKS . . . . .	91
9.20.1.59 USE_HAL_CRYP_REGISTER_CALLBACKS . . . . .	91
9.20.1.60 USE_HAL_DAC_REGISTER_CALLBACKS . . . . .	91
9.20.1.61 USE_HAL_DCMI_REGISTER_CALLBACKS . . . . .	91
9.20.1.62 USE_HAL_DFSDM_REGISTER_CALLBACKS . . . . .	91
9.20.1.63 USE_HAL_DMA2D_REGISTER_CALLBACKS . . . . .	91

9.20.1.64 USE_HAL_DSI_REGISTER_CALLBACKS . . . . .	91
9.20.1.65 USE_HAL_ETH_REGISTER_CALLBACKS . . . . .	91
9.20.1.66 USE_HAL_FMPI2C_REGISTER_CALLBACKS . . . . .	91
9.20.1.67 USE_HAL_FMPMBUS_REGISTER_CALLBACKS . . . . .	92
9.20.1.68 USE_HAL_HASH_REGISTER_CALLBACKS . . . . .	92
9.20.1.69 USE_HAL_HCD_REGISTER_CALLBACKS . . . . .	92
9.20.1.70 USE_HAL_I2C_REGISTER_CALLBACKS . . . . .	92
9.20.1.71 USE_HAL_I2S_REGISTER_CALLBACKS . . . . .	92
9.20.1.72 USE_HAL_IRDA_REGISTER_CALLBACKS . . . . .	92
9.20.1.73 USE_HAL_LPTIM_REGISTER_CALLBACKS . . . . .	92
9.20.1.74 USE_HAL_LTDC_REGISTER_CALLBACKS . . . . .	92
9.20.1.75 USE_HAL_MMC_REGISTER_CALLBACKS . . . . .	92
9.20.1.76 USE_HAL_NAND_REGISTER_CALLBACKS . . . . .	92
9.20.1.77 USE_HAL_NOR_REGISTER_CALLBACKS . . . . .	93
9.20.1.78 USE_HAL_PCCARD_REGISTER_CALLBACKS . . . . .	93
9.20.1.79 USE_HAL_PCD_REGISTER_CALLBACKS . . . . .	93
9.20.1.80 USE_HAL_QSPI_REGISTER_CALLBACKS . . . . .	93
9.20.1.81 USE_HAL_RNG_REGISTER_CALLBACKS . . . . .	93
9.20.1.82 USE_HAL_RTC_REGISTER_CALLBACKS . . . . .	93
9.20.1.83 USE_HAL_SAI_REGISTER_CALLBACKS . . . . .	93
9.20.1.84 USE_HAL_SD_REGISTER_CALLBACKS . . . . .	93
9.20.1.85 USE_HAL_SDRAM_REGISTER_CALLBACKS . . . . .	93
9.20.1.86 USE_HAL_SMARTCARD_REGISTER_CALLBACKS . . . . .	93
9.20.1.87 USE_HAL_SMBUS_REGISTER_CALLBACKS . . . . .	94
9.20.1.88 USE_HAL_SPDIFRX_REGISTER_CALLBACKS . . . . .	94
9.20.1.89 USE_HAL_SPI_REGISTER_CALLBACKS . . . . .	94
9.20.1.90 USE_HAL_SRAM_REGISTER_CALLBACKS . . . . .	94
9.20.1.91 USE_HAL_TIM_REGISTER_CALLBACKS . . . . .	94
9.20.1.92 USE_HAL_UART_REGISTER_CALLBACKS . . . . .	94
9.20.1.93 USE_HAL_USART_REGISTER_CALLBACKS . . . . .	94
9.20.1.94 USE_HAL_WWDG_REGISTER_CALLBACKS . . . . .	94
9.20.1.95 USE_RTOS . . . . .	94
9.20.1.96 USE_SPI_CRC . . . . .	94
9.20.1.97 VDD_VALUE . . . . .	95
9.21 Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/stm32f4xx_hal_conf.h File Reference . . .	95
9.21.1 Macro Definition Documentation . . . . .	97
9.21.1.1 assert_param . . . . .	97
9.21.1.2 DATA_CACHE_ENABLE . . . . .	97
9.21.1.3 DP83848_PHY_ADDRESS . . . . .	97
9.21.1.4 ETH_RX_BUF_SIZE . . . . .	97
9.21.1.5 ETH_RXBUFNB . . . . .	98
9.21.1.6 ETH_TX_BUF_SIZE . . . . .	98

9.21.1.7 ETH_TXBUFNB . . . . .	98
9.21.1.8 EXTERNAL_CLOCK_VALUE . . . . .	98
9.21.1.9 HAL_CORTEX_MODULE_ENABLED . . . . .	98
9.21.1.10 HAL_DMA_MODULE_ENABLED . . . . .	98
9.21.1.11 HAL_EXTI_MODULE_ENABLED . . . . .	98
9.21.1.12 HAL_FLASH_MODULE_ENABLED . . . . .	98
9.21.1.13 HAL_GPIO_MODULE_ENABLED . . . . .	98
9.21.1.14 HAL_MODULE_ENABLED . . . . .	99
9.21.1.15 HAL_PWR_MODULE_ENABLED . . . . .	99
9.21.1.16 HAL_RCC_MODULE_ENABLED . . . . .	99
9.21.1.17 HAL_TIM_MODULE_ENABLED . . . . .	99
9.21.1.18 HAL_UART_MODULE_ENABLED . . . . .	99
9.21.1.19 HSE_STARTUP_TIMEOUT . . . . .	99
9.21.1.20 HSE_VALUE . . . . .	99
9.21.1.21 HSI_VALUE . . . . .	99
9.21.1.22 INSTRUCTION_CACHE_ENABLE . . . . .	100
9.21.1.23 LSE_STARTUP_TIMEOUT . . . . .	100
9.21.1.24 LSE_VALUE . . . . .	100
9.21.1.25 LSI_VALUE . . . . .	100
9.21.1.26 MAC_ADDR0 . . . . .	100
9.21.1.27 MAC_ADDR1 . . . . .	100
9.21.1.28 MAC_ADDR2 . . . . .	100
9.21.1.29 MAC_ADDR3 . . . . .	100
9.21.1.30 MAC_ADDR4 . . . . .	101
9.21.1.31 MAC_ADDR5 . . . . .	101
9.21.1.32 PHY_AUTONEGO_COMPLETE . . . . .	101
9.21.1.33 PHY_AUTONEGOTIATION . . . . .	101
9.21.1.34 PHY_BCR . . . . .	101
9.21.1.35 PHY_BSR . . . . .	101
9.21.1.36 PHY_CONFIG_DELAY . . . . .	101
9.21.1.37 PHY_DUPLEX_STATUS . . . . .	101
9.21.1.38 PHY_FULLDUPLEX_100M . . . . .	102
9.21.1.39 PHY_FULLDUPLEX_10M . . . . .	102
9.21.1.40 PHY_HALFDUPLEX_100M . . . . .	102
9.21.1.41 PHY_HALFDUPLEX_10M . . . . .	102
9.21.1.42 PHY_ISOLATE . . . . .	102
9.21.1.43 PHY_JABBER_DETECTION . . . . .	102
9.21.1.44 PHY_LINKED_STATUS . . . . .	102
9.21.1.45 PHY_LOOPBACK . . . . .	103
9.21.1.46 PHY_POWERDOWN . . . . .	103
9.21.1.47 PHY_READ_TO . . . . .	103
9.21.1.48 PHY_RESET . . . . .	103

9.21.1.49 PHY_RESET_DELAY . . . . .	103
9.21.1.50 PHY_RESTART_AUTONEGOTIATION . . . . .	103
9.21.1.51 PHY_SPEED_STATUS . . . . .	103
9.21.1.52 PHY_SR . . . . .	103
9.21.1.53 PHY_WRITE_TO . . . . .	104
9.21.1.54 PREFETCH_ENABLE . . . . .	104
9.21.1.55 TICK_INT_PRIORITY . . . . .	104
9.21.1.56 USE_HAL_ADC_REGISTER_CALLBACKS . . . . .	104
9.21.1.57 USE_HAL_CAN_REGISTER_CALLBACKS . . . . .	104
9.21.1.58 USE_HAL_CEC_REGISTER_CALLBACKS . . . . .	104
9.21.1.59 USE_HAL_CRYP_REGISTER_CALLBACKS . . . . .	104
9.21.1.60 USE_HAL_DAC_REGISTER_CALLBACKS . . . . .	104
9.21.1.61 USE_HAL_DCMI_REGISTER_CALLBACKS . . . . .	104
9.21.1.62 USE_HAL_DFSDM_REGISTER_CALLBACKS . . . . .	104
9.21.1.63 USE_HAL_DMA2D_REGISTER_CALLBACKS . . . . .	105
9.21.1.64 USE_HAL_DSI_REGISTER_CALLBACKS . . . . .	105
9.21.1.65 USE_HAL_ETH_REGISTER_CALLBACKS . . . . .	105
9.21.1.66 USE_HAL_FMPI2C_REGISTER_CALLBACKS . . . . .	105
9.21.1.67 USE_HAL_FMPMBUS_REGISTER_CALLBACKS . . . . .	105
9.21.1.68 USE_HAL_HASH_REGISTER_CALLBACKS . . . . .	105
9.21.1.69 USE_HAL_HCD_REGISTER_CALLBACKS . . . . .	105
9.21.1.70 USE_HAL_I2C_REGISTER_CALLBACKS . . . . .	105
9.21.1.71 USE_HAL_I2S_REGISTER_CALLBACKS . . . . .	105
9.21.1.72 USE_HAL_IRDA_REGISTER_CALLBACKS . . . . .	105
9.21.1.73 USE_HAL_LPTIM_REGISTER_CALLBACKS . . . . .	106
9.21.1.74 USE_HAL_LTDC_REGISTER_CALLBACKS . . . . .	106
9.21.1.75 USE_HAL_MMC_REGISTER_CALLBACKS . . . . .	106
9.21.1.76 USE_HAL_NAND_REGISTER_CALLBACKS . . . . .	106
9.21.1.77 USE_HAL_NOR_REGISTER_CALLBACKS . . . . .	106
9.21.1.78 USE_HAL_PCCARD_REGISTER_CALLBACKS . . . . .	106
9.21.1.79 USE_HAL_PCD_REGISTER_CALLBACKS . . . . .	106
9.21.1.80 USE_HAL_QSPI_REGISTER_CALLBACKS . . . . .	106
9.21.1.81 USE_HAL_RNG_REGISTER_CALLBACKS . . . . .	106
9.21.1.82 USE_HAL_RTC_REGISTER_CALLBACKS . . . . .	106
9.21.1.83 USE_HAL_SAI_REGISTER_CALLBACKS . . . . .	107
9.21.1.84 USE_HAL_SD_REGISTER_CALLBACKS . . . . .	107
9.21.1.85 USE_HAL_SDRAM_REGISTER_CALLBACKS . . . . .	107
9.21.1.86 USE_HAL_SMARTCARD_REGISTER_CALLBACKS . . . . .	107
9.21.1.87 USE_HAL_SMBUS_REGISTER_CALLBACKS . . . . .	107
9.21.1.88 USE_HAL_SPDIFRX_REGISTER_CALLBACKS . . . . .	107
9.21.1.89 USE_HAL_SPI_REGISTER_CALLBACKS . . . . .	107
9.21.1.90 USE_HAL_SRAM_REGISTER_CALLBACKS . . . . .	107

9.21.1.91 USE_HAL_TIM_REGISTER_CALLBACKS . . . . .	107
9.21.1.92 USE_HAL_UART_REGISTER_CALLBACKS . . . . .	107
9.21.1.93 USE_HAL_USART_REGISTER_CALLBACKS . . . . .	108
9.21.1.94 USE_HAL_WWDG_REGISTER_CALLBACKS . . . . .	108
9.21.1.95 USE_RTOS . . . . .	108
9.21.1.96 USE_SPI_CRC . . . . .	108
9.21.1.97 VDD_VALUE . . . . .	108
9.22 Ball_Launcher_Controller/Core/Inc/stm32f4xx_hal_conf.h File Reference . . . . .	108
9.22.1 Macro Definition Documentation . . . . .	111
9.22.1.1 assert_param . . . . .	111
9.22.1.2 DATA_CACHE_ENABLE . . . . .	111
9.22.1.3 DP83848_PHY_ADDRESS . . . . .	111
9.22.1.4 ETH_RX_BUF_SIZE . . . . .	111
9.22.1.5 ETH_RXBUFNB . . . . .	111
9.22.1.6 ETH_TX_BUF_SIZE . . . . .	111
9.22.1.7 ETH_TXBUFNB . . . . .	111
9.22.1.8 EXTERNAL_CLOCK_VALUE . . . . .	111
9.22.1.9 HAL_CORTEX_MODULE_ENABLED . . . . .	112
9.22.1.10 HAL_DMA_MODULE_ENABLED . . . . .	112
9.22.1.11 HAL_EXTI_MODULE_ENABLED . . . . .	112
9.22.1.12 HAL_FLASH_MODULE_ENABLED . . . . .	112
9.22.1.13 HAL_GPIO_MODULE_ENABLED . . . . .	112
9.22.1.14 HAL_I2C_MODULE_ENABLED . . . . .	112
9.22.1.15 HAL_MODULE_ENABLED . . . . .	112
9.22.1.16 HAL_PWR_MODULE_ENABLED . . . . .	112
9.22.1.17 HAL_RCC_MODULE_ENABLED . . . . .	112
9.22.1.18 HAL_UART_MODULE_ENABLED . . . . .	112
9.22.1.19 HSE_STARTUP_TIMEOUT . . . . .	113
9.22.1.20 HSE_VALUE . . . . .	113
9.22.1.21 HSI_VALUE . . . . .	113
9.22.1.22 INSTRUCTION_CACHE_ENABLE . . . . .	113
9.22.1.23 LSE_STARTUP_TIMEOUT . . . . .	113
9.22.1.24 LSE_VALUE . . . . .	113
9.22.1.25 LSI_VALUE . . . . .	114
9.22.1.26 MAC_ADDR0 . . . . .	114
9.22.1.27 MAC_ADDR1 . . . . .	114
9.22.1.28 MAC_ADDR2 . . . . .	114
9.22.1.29 MAC_ADDR3 . . . . .	114
9.22.1.30 MAC_ADDR4 . . . . .	114
9.22.1.31 MAC_ADDR5 . . . . .	114
9.22.1.32 PHY_AUTONEGO_COMPLETE . . . . .	114
9.22.1.33 PHY_AUTONEGOTIATION . . . . .	115



9.22.1.34 PHY_BCR . . . . .	115
9.22.1.35 PHY_BSR . . . . .	115
9.22.1.36 PHY_CONFIG_DELAY . . . . .	115
9.22.1.37 PHY_DUPLEX_STATUS . . . . .	115
9.22.1.38 PHY_FULLDUPLEX_100M . . . . .	115
9.22.1.39 PHY_FULLDUPLEX_10M . . . . .	115
9.22.1.40 PHY_HALFDUPLEX_100M . . . . .	116
9.22.1.41 PHY_HALFDUPLEX_10M . . . . .	116
9.22.1.42 PHY_ISOLATE . . . . .	116
9.22.1.43 PHY_JABBER_DETECTION . . . . .	116
9.22.1.44 PHY_LINKED_STATUS . . . . .	116
9.22.1.45 PHY_LOOPBACK . . . . .	116
9.22.1.46 PHY_POWERDOWN . . . . .	116
9.22.1.47 PHY_READ_TO . . . . .	117
9.22.1.48 PHY_RESET . . . . .	117
9.22.1.49 PHY_RESET_DELAY . . . . .	117
9.22.1.50 PHY_RESTART_AUTONEGOTIATION . . . . .	117
9.22.1.51 PHY_SPEED_STATUS . . . . .	117
9.22.1.52 PHY_SR . . . . .	117
9.22.1.53 PHY_WRITE_TO . . . . .	117
9.22.1.54 PREFETCH_ENABLE . . . . .	117
9.22.1.55 TICK_INT_PRIORITY . . . . .	118
9.22.1.56 USE_HAL_ADC_REGISTER_CALLBACKS . . . . .	118
9.22.1.57 USE_HAL_CAN_REGISTER_CALLBACKS . . . . .	118
9.22.1.58 USE_HAL_CEC_REGISTER_CALLBACKS . . . . .	118
9.22.1.59 USE_HAL_Cryp_REGISTER_CALLBACKS . . . . .	118
9.22.1.60 USE_HAL_DAC_REGISTER_CALLBACKS . . . . .	118
9.22.1.61 USE_HAL_DCMI_REGISTER_CALLBACKS . . . . .	118
9.22.1.62 USE_HAL_DFSDM_REGISTER_CALLBACKS . . . . .	118
9.22.1.63 USE_HAL_DMA2D_REGISTER_CALLBACKS . . . . .	118
9.22.1.64 USE_HAL_DSI_REGISTER_CALLBACKS . . . . .	118
9.22.1.65 USE_HAL_ETH_REGISTER_CALLBACKS . . . . .	119
9.22.1.66 USE_HAL_FMPI2C_REGISTER_CALLBACKS . . . . .	119
9.22.1.67 USE_HAL_FMPMBUS_REGISTER_CALLBACKS . . . . .	119
9.22.1.68 USE_HAL_HASH_REGISTER_CALLBACKS . . . . .	119
9.22.1.69 USE_HAL_HCD_REGISTER_CALLBACKS . . . . .	119
9.22.1.70 USE_HAL_I2C_REGISTER_CALLBACKS . . . . .	119
9.22.1.71 USE_HAL_I2S_REGISTER_CALLBACKS . . . . .	119
9.22.1.72 USE_HAL_IRDA_REGISTER_CALLBACKS . . . . .	119
9.22.1.73 USE_HAL_LPTIM_REGISTER_CALLBACKS . . . . .	119
9.22.1.74 USE_HAL_LTDC_REGISTER_CALLBACKS . . . . .	119
9.22.1.75 USE_HAL_MMC_REGISTER_CALLBACKS . . . . .	120

9.22.1.76 USE_HAL_NAND_REGISTER_CALLBACKS . . . . .	120
9.22.1.77 USE_HAL_NOR_REGISTER_CALLBACKS . . . . .	120
9.22.1.78 USE_HAL_PCCARD_REGISTER_CALLBACKS . . . . .	120
9.22.1.79 USE_HAL_PCD_REGISTER_CALLBACKS . . . . .	120
9.22.1.80 USE_HAL_QSPI_REGISTER_CALLBACKS . . . . .	120
9.22.1.81 USE_HAL_RNG_REGISTER_CALLBACKS . . . . .	120
9.22.1.82 USE_HAL_RTC_REGISTER_CALLBACKS . . . . .	120
9.22.1.83 USE_HAL_SAI_REGISTER_CALLBACKS . . . . .	120
9.22.1.84 USE_HAL_SD_REGISTER_CALLBACKS . . . . .	120
9.22.1.85 USE_HAL_SDRAM_REGISTER_CALLBACKS . . . . .	121
9.22.1.86 USE_HAL_SMARTCARD_REGISTER_CALLBACKS . . . . .	121
9.22.1.87 USE_HAL_SMBUS_REGISTER_CALLBACKS . . . . .	121
9.22.1.88 USE_HAL_SPDIFRX_REGISTER_CALLBACKS . . . . .	121
9.22.1.89 USE_HAL_SPI_REGISTER_CALLBACKS . . . . .	121
9.22.1.90 USE_HAL_SRAM_REGISTER_CALLBACKS . . . . .	121
9.22.1.91 USE_HAL_TIM_REGISTER_CALLBACKS . . . . .	121
9.22.1.92 USE_HAL_UART_REGISTER_CALLBACKS . . . . .	121
9.22.1.93 USE_HAL_USART_REGISTER_CALLBACKS . . . . .	121
9.22.1.94 USE_HAL_WWDG_REGISTER_CALLBACKS . . . . .	121
9.22.1.95 USE_RTOS . . . . .	122
9.22.1.96 USE_SPI_CRC . . . . .	122
9.22.1.97 VDD_VALUE . . . . .	122
9.23 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/stm32f4xx_it.h File Reference . . . . .	122
9.23.1 Detailed Description . . . . .	123
9.23.2 Function Documentation . . . . .	123
9.23.2.1 BusFault_Handler() . . . . .	123
9.23.2.2 DebugMon_Handler() . . . . .	123
9.23.2.3 HardFault_Handler() . . . . .	123
9.23.2.4 MemManage_Handler() . . . . .	123
9.23.2.5 NMI_Handler() . . . . .	123
9.23.2.6 PendSV_Handler() . . . . .	124
9.23.2.7 SVC_Handler() . . . . .	124
9.23.2.8 SysTick_Handler() . . . . .	124
9.23.2.9 UsageFault_Handler() . . . . .	124
9.24 Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/stm32f4xx_it.h File Reference . . . . .	124
9.24.1 Detailed Description . . . . .	125
9.24.2 Function Documentation . . . . .	125
9.24.2.1 BusFault_Handler() . . . . .	125
9.24.2.2 DebugMon_Handler() . . . . .	125
9.24.2.3 HardFault_Handler() . . . . .	125
9.24.2.4 MemManage_Handler() . . . . .	125
9.24.2.5 NMI_Handler() . . . . .	125

9.24.2.6 PendSV_Handler()	126
9.24.2.7 SVC_Handler()	126
9.24.2.8 SysTick_Handler()	126
9.24.2.9 TIM4_IRQHandler()	126
9.24.2.10 UsageFault_Handler()	126
9.24.2.11 USART1_IRQHandler()	126
9.25 Ball_Launcher_Controller/Core/Inc/stm32f4xx_it.h File Reference	126
9.25.1 Detailed Description	127
9.25.2 Function Documentation	127
9.25.2.1 BusFault_Handler()	127
9.25.2.2 DebugMon_Handler()	127
9.25.2.3 HardFault_Handler()	128
9.25.2.4 MemManage_Handler()	128
9.25.2.5 NMI_Handler()	128
9.25.2.6 PendSV_Handler()	128
9.25.2.7 SVC_Handler()	128
9.25.2.8 SysTick_Handler()	128
9.25.2.9 UsageFault_Handler()	128
9.26 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/main.c File Reference	129
9.26.1 Detailed Description	129
9.26.2 Function Documentation	129
9.26.2.1 Error_Handler()	129
9.26.2.2 main()	130
9.26.2.3 SystemClock_Config()	130
9.26.3 Variable Documentation	130
9.26.3.1 buffer	130
9.26.3.2 gX	130
9.26.3.3 gY	130
9.26.3.4 gZ	130
9.26.3.5 hi2c1	131
9.26.3.6 huart1	131
9.26.3.7 huart2	131
9.26.3.8 move	131
9.26.3.9 shot	131
9.26.3.10 stat	131
9.27 Ball_Launcher_Controller/Core/Src/main.c File Reference	131
9.27.1 Detailed Description	132
9.27.2 Function Documentation	132
9.27.2.1 Error_Handler()	132
9.27.2.2 main()	132
9.27.2.3 SystemClock_Config()	132
9.27.3 Variable Documentation	134

9.27.3.1 buffer	134
9.27.3.2 gX	134
9.27.3.3 gY	134
9.27.3.4 gZ	134
9.27.3.5 hi2c1	134
9.27.3.6 huart1	134
9.27.3.7 huart2	134
9.27.3.8 move	134
9.27.3.9 shot	135
9.27.3.10 stat	135
9.28 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/mpu6050.c File Reference	135
9.28.1 Detailed Description	135
9.28.2 Function Documentation	135
9.28.2.1 mpu6050_calibrate()	135
9.28.2.2 mpu6050_get_gX()	136
9.28.2.3 mpu6050_get_gY()	136
9.28.2.4 mpu6050_get_gZ()	136
9.28.2.5 mpu6050_init()	137
9.28.2.6 mpu6050_update()	137
9.29 Ball_Launcher_Controller/Core/Src/mpu6050.c File Reference	137
9.29.1 Detailed Description	138
9.29.2 Function Documentation	138
9.29.2.1 mpu6050_calibrate()	138
9.29.2.2 mpu6050_get_gX()	138
9.29.2.3 mpu6050_get_gY()	138
9.29.2.4 mpu6050_get_gZ()	139
9.29.2.5 mpu6050_init()	139
9.29.2.6 mpu6050_update()	139
9.30 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/stm32f4xx_hal_msp.c File Reference	140
9.30.1 Detailed Description	140
9.30.2 Function Documentation	140
9.30.2.1 HAL_I2C_MspDeInit()	140
9.30.2.2 HAL_I2C_MspInit()	141
9.30.2.3 HAL_MspInit()	141
9.30.2.4 HAL_UART_MspDeInit()	141
9.30.2.5 HAL_UART_MspInit()	142
9.31 Ball_Launcher_Controller/Core/Src/stm32f4xx_hal_msp.c File Reference	142
9.31.1 Detailed Description	143
9.31.2 Function Documentation	143
9.31.2.1 HAL_I2C_MspDeInit()	143
9.31.2.2 HAL_I2C_MspInit()	143
9.31.2.3 HAL_MspInit()	144

9.31.2.4 HAL_UART_MspDeInit()	144
9.31.2.5 HAL_UART_MspInit()	144
9.32 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/stm32f4xx_it.c File Reference	145
9.32.1 Detailed Description	145
9.32.2 Function Documentation	145
9.32.2.1 BusFault_Handler()	145
9.32.2.2 DebugMon_Handler()	146
9.32.2.3 HardFault_Handler()	146
9.32.2.4 MemManage_Handler()	146
9.32.2.5 NMI_Handler()	146
9.32.2.6 PendSV_Handler()	146
9.32.2.7 SVC_Handler()	146
9.32.2.8 SysTick_Handler()	146
9.32.2.9 UsageFault_Handler()	147
9.33 Ball_Launcher_Controller/Core/Src/stm32f4xx_it.c File Reference	147
9.33.1 Detailed Description	147
9.33.2 Function Documentation	148
9.33.2.1 BusFault_Handler()	148
9.33.2.2 DebugMon_Handler()	148
9.33.2.3 HardFault_Handler()	148
9.33.2.4 MemManage_Handler()	148
9.33.2.5 NMI_Handler()	148
9.33.2.6 PendSV_Handler()	148
9.33.2.7 SVC_Handler()	149
9.33.2.8 SysTick_Handler()	149
9.33.2.9 UsageFault_Handler()	149
9.34 Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/syscalls.c File Reference	149
9.34.1 Detailed Description	150
9.34.2 Function Documentation	150
9.34.2.1 __attribute__()	150
9.34.2.2 __io_getchar()	150
9.34.2.3 __io_putchar()	150
9.34.2.4 _close()	150
9.34.2.5 _execve()	151
9.34.2.6 _exit()	151
9.34.2.7 _fork()	151
9.34.2.8 _fstat()	151
9.34.2.9 _getpid()	151
9.34.2.10 _isatty()	151
9.34.2.11 _kill()	151
9.34.2.12 _link()	151
9.34.2.13 _lseek()	152

9.34.2.14	<a href="#">_open()</a>	152
9.34.2.15	<a href="#">_stat()</a>	152
9.34.2.16	<a href="#">_times()</a>	152
9.34.2.17	<a href="#">_unlink()</a>	152
9.34.2.18	<a href="#">_wait()</a>	152
9.34.2.19	<a href="#">initialise_monitor_handles()</a>	152
9.34.3	<a href="#">Variable Documentation</a>	152
9.34.3.1	<a href="#">environ</a>	152
9.35	<a href="#">Ball_Launcher_Controller/Core/Src/syscalls.c File Reference</a>	153
9.35.1	<a href="#">Detailed Description</a>	153
9.35.2	<a href="#">Function Documentation</a>	154
9.35.2.1	<a href="#">__attribute__()</a>	154
9.35.2.2	<a href="#">__io_getchar()</a>	154
9.35.2.3	<a href="#">__io_putchar()</a>	154
9.35.2.4	<a href="#">_close()</a>	154
9.35.2.5	<a href="#">_execve()</a>	154
9.35.2.6	<a href="#">_exit()</a>	154
9.35.2.7	<a href="#">_fork()</a>	154
9.35.2.8	<a href="#">_fstat()</a>	154
9.35.2.9	<a href="#">_getpid()</a>	155
9.35.2.10	<a href="#">_isatty()</a>	155
9.35.2.11	<a href="#">_kill()</a>	155
9.35.2.12	<a href="#">_link()</a>	155
9.35.2.13	<a href="#">_lseek()</a>	155
9.35.2.14	<a href="#">_open()</a>	155
9.35.2.15	<a href="#">_stat()</a>	155
9.35.2.16	<a href="#">_times()</a>	155
9.35.2.17	<a href="#">_unlink()</a>	156
9.35.2.18	<a href="#">_wait()</a>	156
9.35.2.19	<a href="#">initialise_monitor_handles()</a>	156
9.35.3	<a href="#">Variable Documentation</a>	156
9.35.3.1	<a href="#">environ</a>	156
9.36	<a href="#">Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/systemem.c File Reference</a>	156
9.36.1	<a href="#">Detailed Description</a>	156
9.36.2	<a href="#">Function Documentation</a>	157
9.36.2.1	<a href="#">_sbrk()</a>	157
9.37	<a href="#">Ball_Launcher_Controller/Core/Src/systemem.c File Reference</a>	157
9.37.1	<a href="#">Detailed Description</a>	157
9.37.2	<a href="#">Function Documentation</a>	158
9.37.2.1	<a href="#">_sbrk()</a>	158
9.38	<a href="#">Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/system_stm32f4xx.c File Reference</a>	158
9.38.1	<a href="#">Detailed Description</a>	159

---

9.39 Ball_Launcher_Controller/Core/Src/system_stm32f4xx.c File Reference . . . . .	160
9.39.1 Detailed Description . . . . .	160
9.40 pages/hardware.c File Reference . . . . .	161
9.40.1 Detailed Description . . . . .	161
9.41 pages/mainpage.c File Reference . . . . .	161
9.41.1 Detailed Description . . . . .	161
9.42 pages/software.c File Reference . . . . .	161
9.42.1 Detailed Description . . . . .	161
<b>Index</b>	<b>163</b>





# Chapter 1

## MAIN PAGE

### 1.1 Goals

- Understand the use of wireless protocols like Bluetooth
- Understand electronics considerations and budget management using bill of material
- Use SolidWorks to CAD the external components
- Use Fusion 360 to draw the hardware schematics and draw the PCB
- Use C++ to code in C and understand the different functions of each hardware pin.

### 1.2 Requirements

This project requirements were made in California Polytechnic San Luis Obispo and provided by Professor Charlie Revfrem:

#### 1.2.1 The Electronics Requirements:

- A custom PCB designed around an STM32F411 MCU (or similar) programmed in either C, C++ (or Rust, with permission).
- Two or more actuators, such as motors, driving the machine, actuated by suitable electronics, such as motor drivers.
- Two or more unique sensors.
- Some sort of closed-loop control loop or similarly complex algorithm.
- A wireless controller allowing you to command the bot hands-free or to be used as a wireless e-stop. The controller and receiver will be provided to students for use during ME 507.

### 1.2.2 The Manufacturing Requirements:

- 3D Printing: All custom non-flat parts should likely be manufactured using FDM 3D printing.
- PCB Fabrication: Flat parts can be made to order out of PCB material (fiberglass) along with the PCB controlling your robot. This option provides high precision and rigidity.
- Laser-cutting: Flat parts can be made quickly and accurately on the laser cutter if the parts are of a suitable material; many materials are not safe to cut with a laser.
- Water-jetting: Flat parts not suitable for the laser cutter can be made accurately and quickly using the water-jet.

### 1.2.3 The Safety Requirements:

- If you use a battery, you need to communicate with your instructor before plugging it in or charging it in the lab.
- Your robot must have an emergency stop feature that can be triggered by the provided radio transmitter. For example, releasing the trigger on the transmitter can act as a “dead man’s switch”. The emergency stop must also activate if communication is lost with the wireless controller.
- You may also want to include a way to safely grab your robot if it is mobile in nature, just in case your robot gains sentience and goes after one of the operators.
- Your robot may not injure a human being or, through inaction, allow a human being to come to harm.
- Your robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
- Your robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

## 1.3 Hardware

If you want to learn more about the hardware, click [here](#).

## 1.4 Software

If you want to learn more about the software, click [here](#).

## 1.5 REPOSITORY REFERENCE

All code that will be referenced in this portfolio relate to the project is accessible through <https://github.com/jlam94/ME-507-Portfolio>.

However, you may find it more useful to read through the website exploring the source code.

## 1.6 CONTACT INFO

**Author:** Jonathan Lam

**Email:** [jonathanlam0806@gmail.com](mailto:jonathanlam0806@gmail.com)

**LinkedIn:** [www.linkedin.com/in/jonathanlam2000](http://www.linkedin.com/in/jonathanlam2000)

**Phone Number:** (909) 760-9129

**Major:** Mechanical Engineer at Cal Poly San Luis Obispo

**Date:** May 18, 2024



## Chapter 2

# Hardware Documentation

### 2.1 Introduction

This section provides details about the parts, costs, and how to solve common problems.

### 2.2 Initial Concept

The initial idea was provided by my lab partner Vinh Vo, inspired by his personal project, the cycloidal gearbox. Curious about how we could implement it into a full-motion project, we decided to use an Inertial Measurement Unit (IMU) to control the cycloidal gearbox. To apply our concepts in Mechanical Engineering with Dynamics, we decided it would throw a projectile. Our professor, Charlie Refvem, suggested making it wirelessly controlled, specifically using Bluetooth. With this information, we designed the first concept using an electric schematic for the parts, as seen in the picture.

.

### 2.3 Bill of Materials

As part of the requirements for this project, our team created a bill of materials, which can be seen [here](#). The first tab shows the mechanical parts and parts that we got from Amazon, while the second tab shows our electronic components from the DigiKey website. The URLs are in the Bill of Materials, and the CAD files can be downloaded from there.

### 2.4 Schematic

In this section, the basketball launcher eCAD schematics are shown. The custom PCB is detailed in these pages:

#### 2.4.0.1 Custom PCB Schematics

Page	Description
1	Power Related Parts
2	Microcontroller
3	Motor Related Parts
4	Additional Components and ESC

- 
- 
- 
- 

#### 2.4.0.2 Controller Glove PCB

These images show the PCB for the controller glove and how it is connected to the IMU:

- 

## 2.5 Common Challenges

This section shows some common challenges we encountered and how to troubleshoot them:

- **Problem:** Solder sticking to the board causing parts to short.
  - **Solution:** Look carefully for soldering blobs, and do not use too much soldering wire; blobs are caused by excess solder.
- **Problem:** The library does not show up in the Fusion 360 eDrawings.
  - **Solution:**
    1. Download the part with the STEP file and the library from DigiKey.
    2. Open Fusion 360.
    3. Create a new folder and a new electronics library.
    4. Open Library Manager and import libraries from the local disk.
    5. Right-click the library and click "Edit Fusion 360 copy".
    6. Click Footprint and move the name and values.
    7. Right-click and select 3D sketch.
    8. Open the STEP file and drag it into the 3D drawing; it should be linked after you save.
- **Problem:** The limit switch does not trigger the GPIO.
  - **Solution:** This can be due to the LED in the limit switch that is included in the product; it reduces the voltage output. It is recommended to short the LED and use the limit switch.
- **Problem:** In Fusion 360, the routing error "PCB out of Sync" prevents routing.
  - **Solution:** Push the schematic, then the footprint, so it updates in the PCB routing.
- **Problem:** In Fusion 360, the routing error "Overlap" occurs in the microcontroller ground vias.
  - **Solution:** Click on the part and select "Approved" in the DRC.
- **Problem:** How do you know how much trace width you need?

- **Solution:** Use a trace width calculator. I recommend [this one](#).
- **Problem:** How do you know if the LED or parts are connected on your PCB correctly so they don't accidentally short circuit?
  - **Solution:** Use a multimeter and check for continuity; if it is connected correctly, it should beep.
- **Problem:** The brushless DC motor does not launch the ball far enough.
  - **Solution:** The issue is that the rubber band that contacts the ball was press-fitted, and at high speeds, the rubber dilates, blocking the ball from launching. The solution is to glue the rubber into the CAD wheel. [Here's a video demonstrating the issue](#).





## Chapter 3

# Software Documentation

### 3.1 Introduction

The main goal of our code is the process of communication between Bluetooth. More specifically, the communication between our Inertia Measurement Unit (IMU). Therefore, our code reads the values from the IMU and then prints them into a UART through Bluetooth and communicates them to the master. Additionally, our code needed to run in a Finite State Machine (FSM) so it doesn't interrupt our code and gives orders to the other motors and the brushless motors to shoot the ball.

### 3.2 Finite State Machine

- 
- 

### 3.3 Code

- It is provided in this [manual](#).



# Chapter 4

## Topic Index

### 4.1 Topics

Here is a list of all topics with brief descriptions:

CMSIS . . . . .	17
Stm32f4xx_system . . . . .	17
STM32F4xx_System_Private_Includes . . . . .	17
STM32F4xx_System_Private_TypesDefinitions . . . . .	19
STM32F4xx_System_Private_Defines . . . . .	19
STM32F4xx_System_Private_Macros . . . . .	19
STM32F4xx_System_Private_Variables . . . . .	19
STM32F4xx_System_Private_FunctionPrototypes . . . . .	20
STM32F4xx_System_Private_Functions . . . . .	20



## Chapter 5

# Data Structure Index

### 5.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">D4215X</a>	Structure to define a D4215 motor with timer, channel, and speed . . . . .	<a href="#">23</a>
<a href="#">LEDX</a>	Structure to define an LED with GPIO port and pin . . . . .	<a href="#">24</a>
<a href="#">MPU6050</a>	Struct representing a mpu6050 imu sensor . . . . .	<a href="#">25</a>
<a href="#">RadioX</a>	Structure to define a radio driver with timer, channels, and pulse width parameters . . . . .	<a href="#">26</a>
<a href="#">StepperX</a>	Structure to define a stepper motor driver with GPIO ports and pins . . . . .	<a href="#">28</a>
<a href="#">SwitchX</a>	Structure to define a switch with GPIO port, pin, and status . . . . .	<a href="#">29</a>



## Chapter 6

# File Index

### 6.1 File List

Here is a list of all files with brief descriptions:

Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/main.h	71
Header for main.c file. This file contains the common defines of the application . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/main_C.h	31
Header for main_C.c file. This file contains the common defines of the application . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/mpu6050.h	74
Header for mpu6050.c file. This file contains the common defines of the application . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/stm32f4xx_hal_conf.h	81
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/stm32f4xx_hal_conf_C.h	32
HAL configuration template file. This file should be copied to the application folder and renamed to stm32f4xx_hal_conf_C.h . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/stm32f4xx_it.h	122
This file contains the headers of the interrupt handlers . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/stm32f4xx_it_C.h	46
This file contains the headers of the interrupt handlers . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/main.c	129
Main program body . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/main_C.c	48
Main program body . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/mpu6050.c	135
Implementation of mpu6050 sensor functions . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/stm32f4xx_hal_msp.c	140
This file provides code for the MSP Initialization and de-Initialization codes . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/stm32f4xx_hal_msp_C.c	52
This file provides code for the MSP Initialization and de-Initialization codes . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/stm32f4xx_it.c	145
Interrupt Service Routines . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/stm32f4xx_it_C.c	54
Interrupt Service Routines . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/syscalls.c	149
STM32CubeIDE Minimal System calls file . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/syscalls_C.c	57
STM32CubeIDE Minimal System calls file . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/systemem.c	156
STM32CubeIDE System Memory calls file . . . . .	
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/systemem_C.c	60
STM32CubeIDE System Memory calls file . . . . .	

Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/ <a href="#">system_stm32f4xx.c</a>	
CMSIS Cortex-M4 Device Peripheral Access Layer System Source File	158
Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/ <a href="#">system_stm32f4xx_C.c</a>	
CMSIS Cortex-M4 Device Peripheral Access Layer System Source File	61
Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/ <a href="#">d4215.h</a>	
Header for d4215.c file. This file contains the common defines of the application	62
Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/ <a href="#">led.h</a>	
Header for led.c file. This file contains the common defines of the application	64
Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/ <a href="#">main.h</a>	
Header for main.c file. This file contains the common defines of the application	72
Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/ <a href="#">radio_driver.h</a>	
Header for radio_driver.c file. This file contains the common defines of the application	65
Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/ <a href="#">stepper_driver.h</a>	
Header for stepper_driver.c file. This file contains the common defines of the application	67
Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/ <a href="#">stm32f4xx_hal_conf.h</a>	95
Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/ <a href="#">stm32f4xx_it.h</a>	
This file contains the headers of the interrupt handlers	124
Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/ <a href="#">switch.h</a>	
Header for switch.c file. This file contains the common defines of the application	70
Ball_Launcher_Controller/Core/Inc/ <a href="#">main.h</a>	
Header for main.c file. This file contains the common defines of the application	73
Ball_Launcher_Controller/Core/Inc/ <a href="#">mpu6050.h</a>	
Header for mpu6050.c file. This file contains the common defines of the application	77
Ball_Launcher_Controller/Core/Inc/ <a href="#">stm32f4xx_hal_conf.h</a>	108
Ball_Launcher_Controller/Core/Inc/ <a href="#">stm32f4xx_it.h</a>	
This file contains the headers of the interrupt handlers	126
Ball_Launcher_Controller/Core/Src/ <a href="#">main.c</a>	
Main program body	131
Ball_Launcher_Controller/Core/Src/ <a href="#">mpu6050.c</a>	
Implementation of mpu6050 sensor functions	137
Ball_Launcher_Controller/Core/Src/ <a href="#">stm32f4xx_hal_msp.c</a>	
This file provides code for the MSP Initialization and de-Initialization codes	142
Ball_Launcher_Controller/Core/Src/ <a href="#">stm32f4xx_it.c</a>	
Interrupt Service Routines	147
Ball_Launcher_Controller/Core/Src/ <a href="#">syscalls.c</a>	
STM32CubeIDE Minimal System calls file	153
Ball_Launcher_Controller/Core/Src/ <a href="#">sysmem.c</a>	
STM32CubeIDE System Memory calls file	157
Ball_Launcher_Controller/Core/Src/ <a href="#">system_stm32f4xx.c</a>	
CMSIS Cortex-M4 Device Peripheral Access Layer System Source File	160
pages/ <a href="#">hardware.c</a>	
Documentation for Hardware	161
pages/ <a href="#">mainpage.c</a>	
Main Page Documentation for Project	161
pages/ <a href="#">software.c</a>	
Documentation for Software	161



# Chapter 7

## Topic Documentation

### 7.1 CMSIS

#### Topics

- [Stm32f4xx\\_system](#)

#### 7.1.1 Detailed Description

#### 7.1.2 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](#)

#### 7.1.2.1 Detailed Description

#### 7.1.2.2 STM32F4xx\_System\_Private\_Includes

#### Macros

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

#### 7.1.2.2.1 Detailed Description

#### 7.1.2.2.2 Macro Definition Documentation

##### 7.1.2.2.2.1 HSE\_VALUE [1/3]

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

Default value of the External oscillator in Hz

##### 7.1.2.2.2.2 HSE\_VALUE [2/3]

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

Default value of the External oscillator in Hz

##### 7.1.2.2.2.3 HSE\_VALUE [3/3]

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

Default value of the External oscillator in Hz

##### 7.1.2.2.2.4 HSI\_VALUE [1/3]

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

Value of the Internal oscillator in Hz

##### 7.1.2.2.2.5 HSI\_VALUE [2/3]

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

Value of the Internal oscillator in Hz

##### 7.1.2.2.2.6 HSI\_VALUE [3/3]

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

Value of the Internal oscillator in Hz

### 7.1.2.3 STM32F4xx\_System\_Private\_TypeDefinitions

### 7.1.2.4 STM32F4xx\_System\_Private\_Defines

### 7.1.2.5 STM32F4xx\_System\_Private\_Macros

### 7.1.2.6 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}
- 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}
- 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}

#### 7.1.2.6.1 Detailed Description

#### 7.1.2.6.2 Variable Documentation

##### 7.1.2.6.2.1 AHBPrescTable [1/3]

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

##### 7.1.2.6.2.2 AHBPrescTable [2/3]

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

##### 7.1.2.6.2.3 AHBPrescTable [3/3]

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

##### 7.1.2.6.2.4 APBPrescTable [1/3]

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

##### 7.1.2.6.2.5 APBPrescTable [2/3]

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

**7.1.2.6.2.6 APBPrescTable [3/3]**

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

**7.1.2.6.2.7 SystemCoreClock [1/3]**

```
uint32_t SystemCoreClock = 16000000
```

**7.1.2.6.2.8 SystemCoreClock [2/3]**

```
uint32_t SystemCoreClock = 16000000
```

**7.1.2.6.2.9 SystemCoreClock [3/3]**

```
uint32_t SystemCoreClock = 16000000
```

**7.1.2.7 STM32F4xx\_System\_Private\_FunctionPrototypes****7.1.2.8 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.*

**7.1.2.8.1 Detailed Description****7.1.2.8.2 Function Documentation****7.1.2.8.2.1 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	
------	--

**7.1.2.8.2.2 SystemInit()**

```
void SystemInit (
    void )
```

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

**Parameters**

None	
------	--

**Return values**

None	
------	--



## Chapter 8

# Data Structure Documentation

### 8.1 D4215X Struct Reference

Structure to define a D4215 motor with timer, channel, and speed.

```
#include <d4215.h>
```

#### Data Fields

- TIM\_HandleTypeDef \* [timer](#)
- uint32\_t [channel](#)
- int32\_t [speed](#)

#### 8.1.1 Detailed Description

Structure to define a D4215 motor with timer, channel, and speed.

#### 8.1.2 Field Documentation

##### 8.1.2.1 channel

```
uint32_t channel
```

Timer channel for the motor

##### 8.1.2.2 speed

```
int32_t speed
```

Speed of the motor

### 8.1.2.3 timer

```
TIM_HandleTypeDef* timer
```

Timer handle for the motor

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

- Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/[d4215.h](#)

## 8.2 LEDX Struct Reference

Structure to define an LED with GPIO port and pin.

```
#include <led.h>
```

### Data Fields

- GPIO\_TypeDef \* [GPIOx](#)
- uint16\_t [PIN](#)

### 8.2.1 Detailed Description

Structure to define an LED with GPIO port and pin.

### 8.2.2 Field Documentation

#### 8.2.2.1 GPIOx

```
GPIO_TypeDef* GPIOx
```

GPIO port of the LED

#### 8.2.2.2 PIN

```
uint16_t PIN
```

GPIO pin of the LED

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

- Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/[led.h](#)



## 8.3 MPU6050 Struct Reference

Struct representing a mpu6050 imu sensor.

```
#include <mpu6050.h>
```

### Data Fields

- I2C\_HandleTypeDef \* [hi2c](#)
- HAL\_StatusTypeDef [status](#)
- uint16\_t [addr](#)
- uint16\_t [dt](#)
- int16\_t [gX](#)
- int16\_t [gY](#)
- int16\_t [gZ](#)
- int16\_t [gX\\_offset](#)
- int16\_t [gY\\_offset](#)
- int16\_t [gZ\\_offset](#)

### 8.3.1 Detailed Description

Struct representing a mpu6050 imu sensor.

### 8.3.2 Field Documentation

#### 8.3.2.1 addr

```
uint16_t addr
```

I2C address

#### 8.3.2.2 dt

```
uint16_t dt
```

Delta time

#### 8.3.2.3 gX

```
int16_t gX
```

Gyroscope X axis data

#### 8.3.2.4 gX\_offset

```
int16_t gX_offset
```

Gyroscope X axis offset

### 8.3.2.5 gY

```
int16_t gY
```

Gyroscope Y axis data

### 8.3.2.6 gY\_offset

```
int16_t gY_offset
```

Gyroscope Y axis offset

### 8.3.2.7 gZ

```
int16_t gZ
```

Gyroscope Z axis data

### 8.3.2.8 gZ\_offset

```
int16_t gZ_offset
```

Gyroscope Z axis offset

### 8.3.2.9 hi2c

```
I2C_HandleTypeDef * hi2c
```

I2C handle

### 8.3.2.10 status

```
HAL_StatusTypeDef status
```

HAL status

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

- Ball\_Launcher\_Controller/Core/Inc/[mpu6050.h](#)
- Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/Inc/[mpu6050.h](#)

## 8.4 RadioX Struct Reference

Structure to define a radio driver with timer, channels, and pulse width parameters.

```
#include <radio_driver.h>
```

## Data Fields

- TIM\_HandleTypeDef \* [timer](#)
- uint32\_t [channel1](#)
- uint32\_t [channel2](#)
- uint32\_t [sum1](#)
- uint32\_t [sum2](#)
- uint32\_t [counter1](#)
- uint32\_t [counter2](#)
- double [pulseWidth](#)

### 8.4.1 Detailed Description

Structure to define a radio driver with timer, channels, and pulse width parameters.

### 8.4.2 Field Documentation

#### 8.4.2.1 [channel1](#)

```
uint32_t channel1
```

Timer channel 1

#### 8.4.2.2 [channel2](#)

```
uint32_t channel2
```

Timer channel 2

#### 8.4.2.3 [counter1](#)

```
uint32_t counter1
```

Counter for channel 1

#### 8.4.2.4 [counter2](#)

```
uint32_t counter2
```

Counter for channel 2

#### 8.4.2.5 [pulseWidth](#)

```
double pulseWidth
```

Pulse width

#### 8.4.2.6 sum1

```
uint32_t sum1
```

Sum of pulse widths for channel 1

#### 8.4.2.7 sum2

```
uint32_t sum2
```

Sum of pulse widths for channel 2

#### 8.4.2.8 timer

```
TIM_HandleTypeDef* timer
```

Timer handle for the radio driver

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

- [Ball\\_Launcher\\_Controller/Ball\\_Launcher\\_Main/Core/Inc/radio\\_driver.h](#)

## 8.5 StepperX Struct Reference

Structure to define a stepper motor driver with GPIO ports and pins.

```
#include <stepper_driver.h>
```

### Data Fields

- GPIO\_TypeDef \* [GPIOx](#)
- uint16\_t [EN\\_PIN](#)
- uint16\_t [DIR\\_PIN](#)
- uint16\_t [STP\\_PIN](#)

### 8.5.1 Detailed Description

Structure to define a stepper motor driver with GPIO ports and pins.

### 8.5.2 Field Documentation

#### 8.5.2.1 DIR\_PIN

```
uint16_t DIR_PIN
```

Direction pin of the stepper motor driver

### 8.5.2.2 EN\_PIN

```
uint16_t EN_PIN
```

Enable pin of the stepper motor driver

### 8.5.2.3 GPIOx

```
GPIO_TypeDef* GPIOx
```

GPIO port of the stepper motor driver

### 8.5.2.4 STP\_PIN

```
uint16_t STP_PIN
```

Step pin of the stepper motor driver

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

- [Ball\\_Launcher\\_Controller/Ball\\_Launcher\\_Main/Core/Inc/stepper\\_driver.h](#)

## 8.6 SwitchX Struct Reference

Structure to define a switch with GPIO port, pin, and status.

```
#include <switch.h>
```

### Data Fields

- GPIO\_TypeDef \* [GPIOx](#)
- uint16\_t [PIN](#)
- uint16\_t [status](#)

### 8.6.1 Detailed Description

Structure to define a switch with GPIO port, pin, and status.

### 8.6.2 Field Documentation

#### 8.6.2.1 GPIOx

```
GPIO_TypeDef* GPIOx
```

GPIO port of the switch

### 8.6.2.2 PIN

`uint16_t PIN`

GPIO pin of the switch

### 8.6.2.3 status

`uint16_t status`

Status of the switch

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

- `Ball_Launcher_Controller/Ball_Launcher_Main/Core/Inc/switch.h`

# Chapter 9

## File Documentation

### 9.1 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/Inc/main\_↵ \_C.h File Reference

Header for [main\\_C.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.*

#### 9.1.1 Detailed Description

Header for [main\\_C.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.

#### 9.1.2 Function Documentation

##### 9.1.2.1 Error\_Handler()

```
void Error_Handler (  
    void )
```

This function is executed in case of error occurrence.

## Return values

None	
------	--

## 9.2 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/↵ Inc/stm32f4xx\_hal\_conf\_C.h File Reference

HAL configuration template file. This file should be copied to the application folder and renamed to [stm32f4xx\\_hal\\_conf\\_C.h](#).

```
#include "stm32f4xx_hal_rcc.h"
#include "stm32f4xx_hal_gpio.h"
#include "stm32f4xx_hal_exti.h"
#include "stm32f4xx_hal_dma.h"
#include "stm32f4xx_hal_cortex.h"
#include "stm32f4xx_hal_flash.h"
#include "stm32f4xx_hal_i2c.h"
#include "stm32f4xx_hal_pwr.h"
#include "stm32f4xx_hal_uart.h"
```

### Macros

- `#define HAL_MODULE_ENABLED`  
*This is the list of modules to be used in the HAL driver.*
- `#define HAL_I2C_MODULE_ENABLED`
- `#define HAL_UART_MODULE_ENABLED`
- `#define HAL_GPIO_MODULE_ENABLED`
- `#define HAL_EXTI_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 25000000U`  
*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 100U`
- `#define HSI_VALUE ((uint32_t)16000000U)`  
*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 32000U`  
*Internal Low Speed oscillator (LSI) value.*
- `#define LSE_VALUE 32768U`  
*External Low Speed oscillator (LSE) value.*
- `#define LSE_STARTUP_TIMEOUT 5000U`
- `#define EXTERNAL_CLOCK_VALUE 12288000U`  
*External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S\_CKIN pad.*
- `#define VDD_VALUE 3300U`



*This is the HAL system configuration section.*

- #define `TICK_INT_PRIORITY` 15U
- #define `USE_RTOS` 0U
- #define `PREFETCH_ENABLE` 1U
- #define `INSTRUCTION_CACHE_ENABLE` 1U
- #define `DATA_CACHE_ENABLE` 1U
- #define `USE_HAL_ADC_REGISTER_CALLBACKS` 0U /\* ADC register callback disabled \*/
- #define `USE_HAL_CAN_REGISTER_CALLBACKS` 0U /\* CAN register callback disabled \*/
- #define `USE_HAL_CEC_REGISTER_CALLBACKS` 0U /\* CEC register callback disabled \*/
- #define `USE_HAL_Cryp_REGISTER_CALLBACKS` 0U /\* CRYP register callback disabled \*/
- #define `USE_HAL_DAC_REGISTER_CALLBACKS` 0U /\* DAC register callback disabled \*/
- #define `USE_HAL_DCMI_REGISTER_CALLBACKS` 0U /\* DCMI register callback disabled \*/
- #define `USE_HAL_DFSDM_REGISTER_CALLBACKS` 0U /\* DFSDM register callback disabled \*/
- #define `USE_HAL_DMA2D_REGISTER_CALLBACKS` 0U /\* DMA2D register callback disabled \*/
- #define `USE_HAL_DSI_REGISTER_CALLBACKS` 0U /\* DSI register callback disabled \*/
- #define `USE_HAL_ETH_REGISTER_CALLBACKS` 0U /\* ETH register callback disabled \*/
- #define `USE_HAL_HASH_REGISTER_CALLBACKS` 0U /\* HASH register callback disabled \*/
- #define `USE_HAL_HCD_REGISTER_CALLBACKS` 0U /\* HCD register callback disabled \*/
- #define `USE_HAL_I2C_REGISTER_CALLBACKS` 0U /\* I2C register callback disabled \*/
- #define `USE_HAL_FMPI2C_REGISTER_CALLBACKS` 0U /\* FMPI2C register callback disabled \*/
- #define `USE_HAL_FMPMBUS_REGISTER_CALLBACKS` 0U /\* FMPSMBUS register callback disabled \*/
- #define `USE_HAL_I2S_REGISTER_CALLBACKS` 0U /\* I2S register callback disabled \*/
- #define `USE_HAL_IRDA_REGISTER_CALLBACKS` 0U /\* IRDA register callback disabled \*/
- #define `USE_HAL_LPTIM_REGISTER_CALLBACKS` 0U /\* LPTIM register callback disabled \*/
- #define `USE_HAL_LTDC_REGISTER_CALLBACKS` 0U /\* LTDC register callback disabled \*/
- #define `USE_HAL_MMC_REGISTER_CALLBACKS` 0U /\* MMC register callback disabled \*/
- #define `USE_HAL_NAND_REGISTER_CALLBACKS` 0U /\* NAND register callback disabled \*/
- #define `USE_HAL_NOR_REGISTER_CALLBACKS` 0U /\* NOR register callback disabled \*/
- #define `USE_HAL_PCCARD_REGISTER_CALLBACKS` 0U /\* PCCARD register callback disabled \*/
- #define `USE_HAL_PCD_REGISTER_CALLBACKS` 0U /\* PCD register callback disabled \*/
- #define `USE_HAL_QSPI_REGISTER_CALLBACKS` 0U /\* QSPI register callback disabled \*/
- #define `USE_HAL_RNG_REGISTER_CALLBACKS` 0U /\* RNG register callback disabled \*/
- #define `USE_HAL_RTC_REGISTER_CALLBACKS` 0U /\* RTC register callback disabled \*/
- #define `USE_HAL_SAI_REGISTER_CALLBACKS` 0U /\* SAI register callback disabled \*/
- #define `USE_HAL_SD_REGISTER_CALLBACKS` 0U /\* SD register callback disabled \*/
- #define `USE_HAL_SMARTCARD_REGISTER_CALLBACKS` 0U /\* SMARTCARD register callback disabled \*/
- #define `USE_HAL_SDRAM_REGISTER_CALLBACKS` 0U /\* SDRAM register callback disabled \*/
- #define `USE_HAL_SRAM_REGISTER_CALLBACKS` 0U /\* SRAM register callback disabled \*/
- #define `USE_HAL_SPDIFRX_REGISTER_CALLBACKS` 0U /\* SPDIFRX register callback disabled \*/
- #define `USE_HAL_SMBUS_REGISTER_CALLBACKS` 0U /\* SMBUS register callback disabled \*/
- #define `USE_HAL_SPI_REGISTER_CALLBACKS` 0U /\* SPI register callback disabled \*/
- #define `USE_HAL_TIM_REGISTER_CALLBACKS` 0U /\* TIM register callback disabled \*/
- #define `USE_HAL_UART_REGISTER_CALLBACKS` 0U /\* UART register callback disabled \*/
- #define `USE_HAL_USART_REGISTER_CALLBACKS` 0U /\* USART register callback disabled \*/
- #define `USE_HAL_WWDG_REGISTER_CALLBACKS` 0U /\* WWDG register callback disabled \*/
- #define `MAC_ADDR0` 2U

*Uncomment the line below to expanse the "assert\_param" macro in the HAL drivers code.*

- #define `MAC_ADDR1` 0U
- #define `MAC_ADDR2` 0U
- #define `MAC_ADDR3` 0U
- #define `MAC_ADDR4` 0U
- #define `MAC_ADDR5` 0U
- #define `ETH_RX_BUF_SIZE` ETH\_MAX\_PACKET\_SIZE /\* `buffer` size for receive \*/

- #define ETH\_TX\_BUF\_SIZE ETH\_MAX\_PACKET\_SIZE /\* buffer size for transmit \*/
- #define ETH\_RXBUFNB 4U /\* 4 Rx buffers of size ETH\_RX\_BUF\_SIZE \*/
- #define ETH\_TXBUFNB 4U /\* 4 Tx buffers of size ETH\_TX\_BUF\_SIZE \*/
- #define DP83848\_PHY\_ADDRESS
- #define PHY\_RESET\_DELAY 0x000000FFU
- #define PHY\_CONFIG\_DELAY 0x00000FFFU
- #define PHY\_READ\_TO 0x0000FFFFU
- #define PHY\_WRITE\_TO 0x0000FFFFU
- #define PHY\_BCR ((uint16\_t)0x0000U)
- #define PHY\_BSR ((uint16\_t)0x0001U)
- #define PHY\_RESET ((uint16\_t)0x8000U)
- #define PHY\_LOOPBACK ((uint16\_t)0x4000U)
- #define PHY\_FULLDUPLEX\_100M ((uint16\_t)0x2100U)
- #define PHY\_HALFDUPLEX\_100M ((uint16\_t)0x2000U)
- #define PHY\_FULLDUPLEX\_10M ((uint16\_t)0x0100U)
- #define PHY\_HALFDUPLEX\_10M ((uint16\_t)0x0000U)
- #define PHY\_AUTONEGOTIATION ((uint16\_t)0x1000U)
- #define PHY\_RESTART\_AUTONEGOTIATION ((uint16\_t)0x0200U)
- #define PHY\_POWERDOWN ((uint16\_t)0x0800U)
- #define PHY\_ISOLATE ((uint16\_t)0x0400U)
- #define PHY\_AUTONEGO\_COMPLETE ((uint16\_t)0x0020U)
- #define PHY\_LINKED\_STATUS ((uint16\_t)0x0004U)
- #define PHY\_JABBER\_DETECTION ((uint16\_t)0x0002U)
- #define PHY\_SR ((uint16\_t))
- #define PHY\_SPEED\_STATUS ((uint16\_t))
- #define PHY\_DUPLEX\_STATUS ((uint16\_t))
- #define USE\_SPI\_CRC 0U
- #define assert\_param(expr) ((void)0U)

*Include module's header file.*

## 9.2.1 Detailed Description

HAL configuration template file. This file should be copied to the application folder and renamed to [stm32f4xx\\_hal\\_conf\\_C.h](#).

### Author

MCD Application Team

### 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.

## 9.2.2 Macro Definition Documentation

### 9.2.2.1 assert\_param

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

Include module's header file.

### 9.2.2.2 DATA\_CACHE\_ENABLE

```
#define DATA_CACHE_ENABLE 1U
```

### 9.2.2.3 DP83848\_PHY\_ADDRESS

```
#define DP83848_PHY_ADDRESS
```

### 9.2.2.4 ETH\_RX\_BUF\_SIZE

```
#define ETH_RX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for receive */
```

### 9.2.2.5 ETH\_RXBUFNB

```
#define ETH_RXBUFNB 4U /* 4 Rx buffers of size ETH_RX_BUF_SIZE */
```

### 9.2.2.6 ETH\_TX\_BUF\_SIZE

```
#define ETH_TX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for transmit */
```

### 9.2.2.7 ETH\_TXBUFNB

```
#define ETH_TXBUFNB 4U /* 4 Tx buffers of size ETH_TX_BUF_SIZE */
```

### 9.2.2.8 EXTERNAL\_CLOCK\_VALUE

```
#define EXTERNAL_CLOCK_VALUE 12288000U
```

External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S\_CKIN pad.

Value of the External audio frequency in Hz

#### 9.2.2.9 HAL\_CORTEX\_MODULE\_ENABLED

```
#define HAL_CORTEX_MODULE_ENABLED
```

#### 9.2.2.10 HAL\_DMA\_MODULE\_ENABLED

```
#define HAL_DMA_MODULE_ENABLED
```

#### 9.2.2.11 HAL\_EXTI\_MODULE\_ENABLED

```
#define HAL_EXTI_MODULE_ENABLED
```

#### 9.2.2.12 HAL\_FLASH\_MODULE\_ENABLED

```
#define HAL_FLASH_MODULE_ENABLED
```

#### 9.2.2.13 HAL\_GPIO\_MODULE\_ENABLED

```
#define HAL_GPIO_MODULE_ENABLED
```

#### 9.2.2.14 HAL\_I2C\_MODULE\_ENABLED

```
#define HAL_I2C_MODULE_ENABLED
```

#### 9.2.2.15 HAL\_MODULE\_ENABLED

```
#define HAL_MODULE_ENABLED
```

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

#### 9.2.2.16 HAL\_PWR\_MODULE\_ENABLED

```
#define HAL_PWR_MODULE_ENABLED
```

#### 9.2.2.17 HAL\_RCC\_MODULE\_ENABLED

```
#define HAL_RCC_MODULE_ENABLED
```

#### 9.2.2.18 HAL\_UART\_MODULE\_ENABLED

```
#define HAL_UART_MODULE_ENABLED
```

### 9.2.2.19 HSE\_STARTUP\_TIMEOUT

```
#define HSE_STARTUP_TIMEOUT 100U
```

Time out for HSE start up, in ms

### 9.2.2.20 HSE\_VALUE

```
#define HSE_VALUE 25000000U
```

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

### 9.2.2.21 HSI\_VALUE

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

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

### 9.2.2.22 INSTRUCTION\_CACHE\_ENABLE

```
#define INSTRUCTION_CACHE_ENABLE 1U
```

### 9.2.2.23 LSE\_STARTUP\_TIMEOUT

```
#define LSE_STARTUP_TIMEOUT 5000U
```

Time out for LSE start up, in ms

### 9.2.2.24 LSE\_VALUE

```
#define LSE_VALUE 32768U
```

External Low Speed oscillator (LSE) value.

< 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 Low Speed oscillator in Hz

#### 9.2.2.25 LSI\_VALUE

```
#define LSI_VALUE 32000U
```

Internal Low Speed oscillator (LSI) value.

LSI Typical Value in Hz

#### 9.2.2.26 MAC\_ADDR0

```
#define MAC_ADDR0 2U
```

Uncomment the line below to expanse the "assert\_param" macro in the HAL drivers code.

#### 9.2.2.27 MAC\_ADDR1

```
#define MAC_ADDR1 0U
```

#### 9.2.2.28 MAC\_ADDR2

```
#define MAC_ADDR2 0U
```

#### 9.2.2.29 MAC\_ADDR3

```
#define MAC_ADDR3 0U
```

#### 9.2.2.30 MAC\_ADDR4

```
#define MAC_ADDR4 0U
```

#### 9.2.2.31 MAC\_ADDR5

```
#define MAC_ADDR5 0U
```

#### 9.2.2.32 PHY\_AUTONEGO\_COMPLETE

```
#define PHY_AUTONEGO_COMPLETE ((uint16_t)0x0020U)
```

Auto-Negotiation process completed

### **9.2.2.33 PHY\_AUTONEGOTIATION**

```
#define PHY_AUTONEGOTIATION ((uint16_t)0x1000U)
```

Enable auto-negotiation function

### **9.2.2.34 PHY\_BCR**

```
#define PHY_BCR ((uint16_t)0x0000U)
```

Transceiver Basic Control Register

### **9.2.2.35 PHY\_BSR**

```
#define PHY_BSR ((uint16_t)0x0001U)
```

Transceiver Basic Status Register

### **9.2.2.36 PHY\_CONFIG\_DELAY**

```
#define PHY_CONFIG_DELAY 0x00000FFFU
```

### **9.2.2.37 PHY\_DUPLEX\_STATUS**

```
#define PHY_DUPLEX_STATUS ((uint16_t))
```

PHY Duplex mask

### **9.2.2.38 PHY\_FULLDUPLEX\_100M**

```
#define PHY_FULLDUPLEX_100M ((uint16_t)0x2100U)
```

Set the full-duplex mode at 100 Mb/s

### **9.2.2.39 PHY\_FULLDUPLEX\_10M**

```
#define PHY_FULLDUPLEX_10M ((uint16_t)0x0100U)
```

Set the full-duplex mode at 10 Mb/s

#### 9.2.2.40 PHY\_HALFDUPLEX\_100M

```
#define PHY_HALFDUPLEX_100M ((uint16_t)0x2000U)
```

Set the half-duplex mode at 100 Mb/s

#### 9.2.2.41 PHY\_HALFDUPLEX\_10M

```
#define PHY_HALFDUPLEX_10M ((uint16_t)0x0000U)
```

Set the half-duplex mode at 10 Mb/s

#### 9.2.2.42 PHY\_ISOLATE

```
#define PHY_ISOLATE ((uint16_t)0x0400U)
```

Isolate PHY from MII

#### 9.2.2.43 PHY\_JABBER\_DETECTION

```
#define PHY_JABBER_DETECTION ((uint16_t)0x0002U)
```

Jabber condition detected

#### 9.2.2.44 PHY\_LINKED\_STATUS

```
#define PHY_LINKED_STATUS ((uint16_t)0x0004U)
```

Valid link established

#### 9.2.2.45 PHY\_LOOPBACK

```
#define PHY_LOOPBACK ((uint16_t)0x4000U)
```

Select loop-back mode

#### 9.2.2.46 PHY\_POWERDOWN

```
#define PHY_POWERDOWN ((uint16_t)0x0800U)
```

Select the power down mode



#### 9.2.2.47 PHY\_READ\_TO

```
#define PHY_READ_TO 0x0000FFFFU
```

#### 9.2.2.48 PHY\_RESET

```
#define PHY_RESET ((uint16_t)0x8000U)
```

PHY Reset

#### 9.2.2.49 PHY\_RESET\_DELAY

```
#define PHY_RESET_DELAY 0x000000FFU
```

#### 9.2.2.50 PHY\_RESTART\_AUTONEGOTIATION

```
#define PHY_RESTART_AUTONEGOTIATION ((uint16_t)0x0200U)
```

Restart auto-negotiation function

#### 9.2.2.51 PHY\_SPEED\_STATUS

```
#define PHY_SPEED_STATUS ((uint16_t))
```

PHY Speed mask

#### 9.2.2.52 PHY\_SR

```
#define PHY_SR ((uint16_t))
```

PHY status register Offset

#### 9.2.2.53 PHY\_WRITE\_TO

```
#define PHY_WRITE_TO 0x0000FFFFU
```

#### 9.2.2.54 PREFETCH\_ENABLE

```
#define PREFETCH_ENABLE 1U
```

#### 9.2.2.55 TICK\_INT\_PRIORITY

```
#define TICK_INT_PRIORITY 15U
```

tick interrupt priority

#### 9.2.2.56 USE\_HAL\_ADC\_REGISTER\_CALLBACKS

```
#define USE_HAL_ADC_REGISTER_CALLBACKS 0U /* ADC register callback disabled */
```

#### 9.2.2.57 USE\_HAL\_CAN\_REGISTER\_CALLBACKS

```
#define USE_HAL_CAN_REGISTER_CALLBACKS 0U /* CAN register callback disabled */
```

#### 9.2.2.58 USE\_HAL\_CEC\_REGISTER\_CALLBACKS

```
#define USE_HAL_CEC_REGISTER_CALLBACKS 0U /* CEC register callback disabled */
```

#### 9.2.2.59 USE\_HAL\_Cryp\_REGISTER\_CALLBACKS

```
#define USE_HAL_Cryp_REGISTER_CALLBACKS 0U /* CRYP register callback disabled */
```

#### 9.2.2.60 USE\_HAL\_DAC\_REGISTER\_CALLBACKS

```
#define USE_HAL_DAC_REGISTER_CALLBACKS 0U /* DAC register callback disabled */
```

#### 9.2.2.61 USE\_HAL\_DCMI\_REGISTER\_CALLBACKS

```
#define USE_HAL_DCMI_REGISTER_CALLBACKS 0U /* DCMI register callback disabled */
```

#### 9.2.2.62 USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS

```
#define USE_HAL_DFSDM_REGISTER_CALLBACKS 0U /* DFSDM register callback disabled */
```

#### 9.2.2.63 USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS

```
#define USE_HAL_DMA2D_REGISTER_CALLBACKS 0U /* DMA2D register callback disabled */
```

#### 9.2.2.64 USE\_HAL\_DSI\_REGISTER\_CALLBACKS

```
#define USE_HAL_DSI_REGISTER_CALLBACKS 0U /* DSI register callback disabled */
```

#### **9.2.2.65 USE\_HAL\_ETH\_REGISTER\_CALLBACKS**

```
#define USE_HAL_ETH_REGISTER_CALLBACKS 0U /* ETH register callback disabled */
```

#### **9.2.2.66 USE\_HAL\_FMPI2C\_REGISTER\_CALLBACKS**

```
#define USE_HAL_FMPI2C_REGISTER_CALLBACKS 0U /* FMPI2C register callback disabled */
```

#### **9.2.2.67 USE\_HAL\_FMPSMBUS\_REGISTER\_CALLBACKS**

```
#define USE_HAL_FMPSMBUS_REGISTER_CALLBACKS 0U /* FMPSMBUS register callback disabled */
```

#### **9.2.2.68 USE\_HAL\_HASH\_REGISTER\_CALLBACKS**

```
#define USE_HAL_HASH_REGISTER_CALLBACKS 0U /* HASH register callback disabled */
```

#### **9.2.2.69 USE\_HAL\_HCD\_REGISTER\_CALLBACKS**

```
#define USE_HAL_HCD_REGISTER_CALLBACKS 0U /* HCD register callback disabled */
```

#### **9.2.2.70 USE\_HAL\_I2C\_REGISTER\_CALLBACKS**

```
#define USE_HAL_I2C_REGISTER_CALLBACKS 0U /* I2C register callback disabled */
```

#### **9.2.2.71 USE\_HAL\_I2S\_REGISTER\_CALLBACKS**

```
#define USE_HAL_I2S_REGISTER_CALLBACKS 0U /* I2S register callback disabled */
```

#### **9.2.2.72 USE\_HAL\_IRDA\_REGISTER\_CALLBACKS**

```
#define USE_HAL_IRDA_REGISTER_CALLBACKS 0U /* IRDA register callback disabled */
```

#### **9.2.2.73 USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS**

```
#define USE_HAL_LPTIM_REGISTER_CALLBACKS 0U /* LPTIM register callback disabled */
```

#### **9.2.2.74 USE\_HAL\_LTDC\_REGISTER\_CALLBACKS**

```
#define USE_HAL_LTDC_REGISTER_CALLBACKS 0U /* LTDC register callback disabled */
```

#### 9.2.2.75 USE\_HAL\_MMC\_REGISTER\_CALLBACKS

```
#define USE_HAL_MMC_REGISTER_CALLBACKS 0U /* MMC register callback disabled */
```

#### 9.2.2.76 USE\_HAL\_NAND\_REGISTER\_CALLBACKS

```
#define USE_HAL_NAND_REGISTER_CALLBACKS 0U /* NAND register callback disabled */
```

#### 9.2.2.77 USE\_HAL\_NOR\_REGISTER\_CALLBACKS

```
#define USE_HAL_NOR_REGISTER_CALLBACKS 0U /* NOR register callback disabled */
```

#### 9.2.2.78 USE\_HAL\_PCCARD\_REGISTER\_CALLBACKS

```
#define USE_HAL_PCCARD_REGISTER_CALLBACKS 0U /* PCCARD register callback disabled */
```

#### 9.2.2.79 USE\_HAL\_PCD\_REGISTER\_CALLBACKS

```
#define USE_HAL_PCD_REGISTER_CALLBACKS 0U /* PCD register callback disabled */
```

#### 9.2.2.80 USE\_HAL\_QSPI\_REGISTER\_CALLBACKS

```
#define USE_HAL_QSPI_REGISTER_CALLBACKS 0U /* QSPI register callback disabled */
```

#### 9.2.2.81 USE\_HAL\_RNG\_REGISTER\_CALLBACKS

```
#define USE_HAL_RNG_REGISTER_CALLBACKS 0U /* RNG register callback disabled */
```

#### 9.2.2.82 USE\_HAL\_RTC\_REGISTER\_CALLBACKS

```
#define USE_HAL_RTC_REGISTER_CALLBACKS 0U /* RTC register callback disabled */
```

#### 9.2.2.83 USE\_HAL\_SAI\_REGISTER\_CALLBACKS

```
#define USE_HAL_SAI_REGISTER_CALLBACKS 0U /* SAI register callback disabled */
```

#### 9.2.2.84 USE\_HAL\_SD\_REGISTER\_CALLBACKS

```
#define USE_HAL_SD_REGISTER_CALLBACKS 0U /* SD register callback disabled */
```

#### **9.2.2.85 USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SDRAM_REGISTER_CALLBACKS 0U /* SDRAM register callback disabled */
```

#### **9.2.2.86 USE\_HAL\_SMARTCARD\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SMARTCARD_REGISTER_CALLBACKS 0U /* SMARTCARD register callback disabled */
```

#### **9.2.2.87 USE\_HAL\_SMBUS\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SMBUS_REGISTER_CALLBACKS 0U /* SMBUS register callback disabled */
```

#### **9.2.2.88 USE\_HAL\_SPDIFRX\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SPDIFRX_REGISTER_CALLBACKS 0U /* SPDIFRX register callback disabled */
```

#### **9.2.2.89 USE\_HAL\_SPI\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SPI_REGISTER_CALLBACKS 0U /* SPI register callback disabled */
```

#### **9.2.2.90 USE\_HAL\_SRAM\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SRAM_REGISTER_CALLBACKS 0U /* SRAM register callback disabled */
```

#### **9.2.2.91 USE\_HAL\_TIM\_REGISTER\_CALLBACKS**

```
#define USE_HAL_TIM_REGISTER_CALLBACKS 0U /* TIM register callback disabled */
```

#### **9.2.2.92 USE\_HAL\_UART\_REGISTER\_CALLBACKS**

```
#define USE_HAL_UART_REGISTER_CALLBACKS 0U /* UART register callback disabled */
```

#### **9.2.2.93 USE\_HAL\_USART\_REGISTER\_CALLBACKS**

```
#define USE_HAL_USART_REGISTER_CALLBACKS 0U /* USART register callback disabled */
```

#### **9.2.2.94 USE\_HAL\_WWDG\_REGISTER\_CALLBACKS**

```
#define USE_HAL_WWDG_REGISTER_CALLBACKS 0U /* WWDG register callback disabled */
```

### 9.2.2.95 USE\_RTOS

```
#define USE_RTOS 0U
```

### 9.2.2.96 USE\_SPI\_CRC

```
#define USE_SPI_CRC 0U
```

### 9.2.2.97 VDD\_VALUE

```
#define VDD_VALUE 3300U
```

This is the HAL system configuration section.

Value of VDD in mv

## 9.3 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/Inc/stm32f4xx\_it\_C.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.*

### 9.3.1 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.

### 9.3.2 Function Documentation

#### 9.3.2.1 BusFault\_Handler()

```
void BusFault_Handler (
    void )
```

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

#### 9.3.2.2 DebugMon\_Handler()

```
void DebugMon_Handler (
    void )
```

This function handles Debug monitor.

#### 9.3.2.3 HardFault\_Handler()

```
void HardFault_Handler (
    void )
```

This function handles Hard fault interrupt.

#### 9.3.2.4 MemManage\_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

#### 9.3.2.5 NMI\_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

#### 9.3.2.6 PendSV\_Handler()

```
void PendSV_Handler (
    void )
```

This function handles Pendable request for system service.

#### 9.3.2.7 SVC\_Handler()

```
void SVC_Handler (
    void )
```

This function handles System service call via SWI instruction.

#### 9.3.2.8 SysTick\_Handler()

```
void SysTick_Handler (
    void )
```

This function handles System tick timer.

#### 9.3.2.9 UsageFault\_Handler()

```
void UsageFault_Handler (
    void )
```

This function handles Undefined instruction or illegal state.

## 9.4 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/Src/main\_↔ \_C.c File Reference

Main program body.

```
#include "main.h"
#include "mpu6050.h"
#include <stdio.h>
#include <string.h>
```

### Functions

- void [SystemClock\\_Config](#) (void)  
*System Clock Configuration.*
- int [main](#) (void)  
*The application entry point.*
- void [Error\\_Handler](#) (void)  
*This function is executed in case of error occurrence.*



## Variables

- I2C\_HandleTypeDef [hi2c1](#)
- UART\_HandleTypeDef [huart1](#)
- UART\_HandleTypeDef [huart2](#)
- int16\_t [gX](#) = 0
- int16\_t [gY](#) = 0
- int16\_t [gZ](#) = 0
- uint16\_t [stat](#) = 0
- uint16\_t [shot](#) = 0
- uint16\_t [move](#) = 0
- uint16\_t [STATE](#) = 0
- uint16\_t [STATE\\_0\\_INIT](#) = 0
- uint16\_t [STATE\\_1\\_ERROR](#) = 1
- uint16\_t [STATE\\_2\\_IMU](#) = 2
- uint16\_t [STATE\\_3\\_BUTTON\\_LED](#) = 3
- uint16\_t [STATE\\_4\\_TRANSFER](#) = 4
- char [buffer](#) [100]

## 9.4.1 Detailed Description

Main program body.

### 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.

## 9.4.2 Function Documentation

### 9.4.2.1 Error\_Handler()

```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

#### Return values

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

### 9.4.2.2 main()

```
int main (
    void )
```

The application entry point.

#### Return values

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

### 9.4.2.3 SystemClock\_Config()

```
void SystemClock_Config (
    void )
```

System Clock Configuration.

#### Return values

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

Configure the main internal regulator output voltage

Initializes the RCC Oscillators according to the specified parameters in the RCC\_OscInitTypeDef structure.

Initializes the CPU, AHB and APB buses clocks

## 9.4.3 Variable Documentation

### 9.4.3.1 buffer

```
char buffer[100]
```

### 9.4.3.2 gX

```
int16_t gX = 0
```

### 9.4.3.3 gY

```
int16_t gY = 0
```

### 9.4.3.4 gZ

```
int16_t gZ = 0
```

### 9.4.3.5 hi2c1

```
I2C_HandleTypeDef hi2c1
```

#### 9.4.3.6 huart1

```
UART_HandleTypeDef huart1
```

#### 9.4.3.7 huart2

```
UART_HandleTypeDef huart2
```

#### 9.4.3.8 move

```
uint16_t move = 0
```

#### 9.4.3.9 shot

```
uint16_t shot = 0
```

#### 9.4.3.10 stat

```
uint16_t stat = 0
```

#### 9.4.3.11 STATE

```
uint16_t STATE = 0
```

#### 9.4.3.12 STATE\_0\_INIT

```
uint16_t STATE_0_INIT = 0
```

#### 9.4.3.13 STATE\_1\_ERROR

```
uint16_t STATE_1_ERROR = 1
```

#### 9.4.3.14 STATE\_2\_IMU

```
uint16_t STATE_2_IMU = 2
```

#### 9.4.3.15 STATE\_3\_BUTTON\_LED

```
uint16_t STATE_3_BUTTON_LED = 3
```

#### 9.4.3.16 STATE\_4\_TRANSFER

```
uint16_t STATE_4_TRANSFER = 4
```

### 9.5 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/↔ Src/stm32f4xx\_hal\_msp\_C.c File Reference

This file provides code for the MSP Initialization and de-Initialization codes.

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

#### Functions

- void [HAL\\_MspInit](#) (void)
- void [HAL\\_I2C\\_MspInit](#) (I2C\_HandleTypeDef \*hi2c)  
*I2C MSP Initialization This function configures the hardware resources used in this example.*
- void [HAL\\_I2C\\_MspDeInit](#) (I2C\_HandleTypeDef \*hi2c)  
*I2C MSP De-Initialization This function freeze the hardware resources used in this example.*
- void [HAL\\_UART\\_MspInit](#) (UART\_HandleTypeDef \*huart)  
*UART MSP Initialization This function configures the hardware resources used in this example.*
- void [HAL\\_UART\\_MspDeInit](#) (UART\_HandleTypeDef \*huart)  
*UART MSP De-Initialization This function freeze the hardware resources used in this example.*

#### 9.5.1 Detailed Description

This file provides code for the MSP Initialization and de-Initialization codes.

#### 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.

#### 9.5.2 Function Documentation

##### 9.5.2.1 HAL\_I2C\_MspDeInit()

```
void HAL_I2C_MspDeInit (
    I2C_HandleTypeDef * hi2c )
```

I2C MSP De-Initialization This function freeze the hardware resources used in this example.

**Parameters**

<i>hi2c</i>	I2C handle pointer
-------------	--------------------

**Return values**

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

I2C1 GPIO Configuration PB6 -----> I2C1\_SCL PB7 -----> I2C1\_SDA

**9.5.2.2 HAL\_I2C\_MspInit()**

```
void HAL_I2C_MspInit (
    I2C_HandleTypeDef * hi2c )
```

I2C MSP Initialization This function configures the hardware resources used in this example.

**Parameters**

<i>hi2c</i>	I2C handle pointer
-------------	--------------------

**Return values**

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

I2C1 GPIO Configuration PB6 -----> I2C1\_SCL PB7 -----> I2C1\_SDA

**9.5.2.3 HAL\_MspInit()**

```
void HAL_MspInit (
    void )
```

Initializes the Global MSP.

**9.5.2.4 HAL\_UART\_MspDeInit()**

```
void HAL_UART_MspDeInit (
    UART_HandleTypeDef * huart )
```

UART MSP De-Initialization This function freeze the hardware resources used in this example.

**Parameters**

<i>huart</i>	UART handle pointer
--------------	---------------------

## Return values

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

USART1 GPIO Configuration PA9 -----> USART1\_TX PA10 -----> USART1\_RX

USART2 GPIO Configuration PA2 -----> USART2\_TX PA3 -----> USART2\_RX

### 9.5.2.5 HAL\_UART\_MspInit()

```
void HAL_UART_MspInit (
    UART_HandleTypeDef * huart )
```

UART MSP Initialization This function configures the hardware resources used in this example.

## Parameters

<i>huart</i>	UART handle pointer
--------------	---------------------

## Return values

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

USART1 GPIO Configuration PA9 -----> USART1\_TX PA10 -----> USART1\_RX

USART2 GPIO Configuration PA2 -----> USART2\_TX PA3 -----> USART2\_RX

## 9.6 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/↵ Src/stm32f4xx\_it\_C.c File Reference

Interrupt Service Routines.

```
#include "main.h"
#include "stm32f4xx_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.*

## 9.6.1 Detailed Description

Interrupt Service Routines.

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.

## 9.6.2 Function Documentation

### 9.6.2.1 BusFault\_Handler()

```
void BusFault_Handler (  
    void )
```

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

### 9.6.2.2 DebugMon\_Handler()

```
void DebugMon_Handler (  
    void )
```

This function handles Debug monitor.

### 9.6.2.3 HardFault\_Handler()

```
void HardFault_Handler (  
    void )
```

This function handles Hard fault interrupt.

#### 9.6.2.4 MemManage\_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

#### 9.6.2.5 NMI\_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

#### 9.6.2.6 PendSV\_Handler()

```
void PendSV_Handler (
    void )
```

This function handles Pendable request for system service.

#### 9.6.2.7 SVC\_Handler()

```
void SVC_Handler (
    void )
```

This function handles System service call via SWI instruction.

#### 9.6.2.8 SysTick\_Handler()

```
void SysTick_Handler (
    void )
```

This function handles System tick timer.

#### 9.6.2.9 UsageFault\_Handler()

```
void UsageFault_Handler (
    void )
```

This function handles Undefined instruction or illegal state.



## 9.7 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/↵ Src/syscalls\_C.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`

### 9.7.1 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) 2020-2024 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.

## 9.7.2 Function Documentation

### 9.7.2.1 `__attribute__()`

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

### 9.7.2.2 `__io_getchar()`

```
int __io_getchar (  
    void ) [extern]
```

### 9.7.2.3 `__io_putchar()`

```
int __io_putchar (  
    int ch ) [extern]
```

### 9.7.2.4 `_close()`

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

### 9.7.2.5 `_execve()`

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

### 9.7.2.6 `_exit()`

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

### 9.7.2.7 `_fork()`

```
int _fork (  
    void )
```

### 9.7.2.8 `_fstat()`

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

### 9.7.2.9 \_getpid()

```
int _getpid (
    void )
```

### 9.7.2.10 \_isatty()

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

### 9.7.2.11 \_kill()

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

### 9.7.2.12 \_link()

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

### 9.7.2.13 \_lseek()

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

### 9.7.2.14 \_open()

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

### 9.7.2.15 \_stat()

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

### 9.7.2.16 \_times()

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

### 9.7.2.17 `_unlink()`

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

### 9.7.2.18 `_wait()`

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

### 9.7.2.19 `initialise_monitor_handles()`

```
void initialise_monitor_handles ( )
```

## 9.7.3 Variable Documentation

### 9.7.3.1 `environ`

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

## 9.8 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/↔ Src/systemem\_C.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*

### 9.8.1 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) 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.

## 9.8.2 Function Documentation

### 9.8.2.1 \_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

## 9.9 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/↵ Src/system\_stm32f4xx\_C.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}

### 9.9.1 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.

## 9.10 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/d4215.h File Reference

Header for d4215.c file. This file contains the common defines of the application.

```
#include <stdio.h>
#include <stdint.h>
#include "stm32f4xx_hal.h"
```

## Data Structures

- struct [D4215X](#)  
*Structure to define a D4215 motor with timer, channel, and speed.*

## Functions

- void [D4215\\_init](#) ([D4215X](#) \*bldcx, TIM\_HandleTypeDef \*timer, uint32\_t channel)  
*Initializes the D4215 motor with the specified timer and channel.*
- void [D4215\\_set](#) ([D4215X](#) \*bldcx, int32\_t speed)  
*Sets the speed of the D4215 motor.*

### 9.10.1 Detailed Description

Header for d4215.c file. This file contains the common defines of the application.

Created on: May 1, 2024

Author: vvinh

### 9.10.2 Function Documentation

#### 9.10.2.1 D4215\_init()

```
void D4215_init (  
    D4215X * bldcx,  
    TIM_HandleTypeDef * timer,  
    uint32_t channel )
```

Initializes the D4215 motor with the specified timer and channel.

##### Parameters

<i>bldcx</i>	Pointer to the <a href="#">D4215X</a> structure
<i>timer</i>	Timer handle
<i>channel</i>	Timer channel

#### 9.10.2.2 D4215\_set()

```
void D4215_set (  
    D4215X * bldcx,  
    int32_t speed )
```

Sets the speed of the D4215 motor.

##### Parameters

<i>bldcx</i>	Pointer to the <a href="#">D4215X</a> structure
<i>speed</i>	Speed of the motor (range: 0 to 100)

## 9.11 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/led.h File Reference

Header for led.c file. This file contains the common defines of the application.

```
#include <stdio.h>
#include <stdint.h>
#include "stm32f4xx_hal.h"
```

### Data Structures

- struct [LEDX](#)  
*Structure to define an LED with GPIO port and pin.*

### Functions

- void [LED\\_init](#) ([LEDX](#) \*LEDx, GPIO\_TypeDef \*GPIO, uint16\_t PIN)  
*Initializes the LED with the specified GPIO port and pin.*
- void [LED\\_on](#) ([LEDX](#) \*LEDx)  
*Turns on the LED.*
- void [LED\\_off](#) ([LEDX](#) \*LEDx)  
*Turns off the LED.*
- void [LED\\_toggle](#) ([LEDX](#) \*LEDx)  
*Toggles the LED state.*

### 9.11.1 Detailed Description

Header for led.c file. This file contains the common defines of the application.

Created on: May 17, 2024

Author: vvinh

### 9.11.2 Function Documentation

#### 9.11.2.1 LED\_init()

```
void LED_init (
    LEDX * LEDx,
    GPIO_TypeDef * GPIO,
    uint16_t PIN )
```

Initializes the LED with the specified GPIO port and pin.

#### Parameters

<i>LEDx</i>	Pointer to the <a href="#">LEDX</a> structure
<i>GPIO</i>	GPIO port
<i>PIN</i>	GPIO pin



### 9.11.2.2 LED\_off()

```
void LED_off (
    LEDX * LEDx )
```

Turns off the LED.

#### Parameters

LEDx	Pointer to the LEDX structure
------	-------------------------------

### 9.11.2.3 LED\_on()

```
void LED_on (
    LEDX * LEDx )
```

Turns on the LED.

#### Parameters

LEDx	Pointer to the LEDX structure
------	-------------------------------

### 9.11.2.4 LED\_toggle()

```
void LED_toggle (
    LEDX * LEDx )
```

Toggles the LED state.

#### Parameters

LEDx	Pointer to the LEDX structure
------	-------------------------------

## 9.12 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/radio\_driver.h File Reference ↩

Header for radio\_driver.c file. This file contains the common defines of the application.

```
#include <stdio.h>
#include <stdint.h>
#include "stm32f4xx_hal.h"
```

### Data Structures

- struct RadioX

*Structure to define a radio driver with timer, channels, and pulse width parameters.*

## Functions

- void `Radio_init` (`RadioX` \*radio, TIM\_HandleTypeDef \*timer, uint32\_t channel1, uint32\_t channel2)  
*Initializes the radio driver with the specified timer and channels.*
- void `capturePulseWidth` (`RadioX` \*radio)  
*Captures the pulse width for the radio driver.*
- void `update` (`RadioX` \*radio, int pw1, int pw2)  
*Updates the radio driver with pulse width values.*
- double `Radio_getPulseWidth` (`RadioX` \*radio)  
*Gets the pulse width of the radio driver.*

### 9.12.1 Detailed Description

Header for radio\_driver.c file. This file contains the common defines of the application.

Created on: May 2, 2024

Author: vvinh

### 9.12.2 Function Documentation

#### 9.12.2.1 capturePulseWidth()

```
void capturePulseWidth (
    RadioX * radio )
```

Captures the pulse width for the radio driver.

##### Parameters

<code>radio</code>	Pointer to the <code>RadioX</code> structure
--------------------	--

#### 9.12.2.2 Radio\_getPulseWidth()

```
double Radio_getPulseWidth (
    RadioX * radio )
```

Gets the pulse width of the radio driver.

##### Parameters

<code>radio</code>	Pointer to the <code>RadioX</code> structure
--------------------	--

##### Returns

double Pulse width value

### 9.12.2.3 Radio\_init()

```
void Radio_init (
    RadioX * radio,
    TIM_HandleTypeDef * timer,
    uint32_t channel1,
    uint32_t channel2 )
```

Initializes the radio driver with the specified timer and channels.

#### Parameters

<i>radio</i>	Pointer to the <a href="#">RadioX</a> structure
<i>timer</i>	Timer handle
<i>channel1</i>	Timer channel 1
<i>channel2</i>	Timer channel 2

### 9.12.2.4 update()

```
void update (
    RadioX * radio,
    int pw1,
    int pw2 )
```

Updates the radio driver with pulse width values.

#### Parameters

<i>radio</i>	Pointer to the <a href="#">RadioX</a> structure
<i>pw1</i>	Pulse width value for channel 1
<i>pw2</i>	Pulse width value for channel 2

## 9.13 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/stepper\_driver.h File Reference

Header for stepper\_driver.c file. This file contains the common defines of the application.

```
#include <stdio.h>
#include <stdint.h>
#include "stm32f4xx_hal.h"
```

### Data Structures

- struct [StepperX](#)

*Structure to define a stepper motor driver with GPIO ports and pins.*

## Functions

- void [Stepper\\_init](#) ([StepperX](#) \*stepperx, GPIO\_TypeDef \*GPIO, uint16\_t EN\_PIN, uint16\_t DIR\_PIN, uint16\_t STP\_PIN)  
*Initializes the stepper motor driver with the specified GPIO port and pins.*
- void [Stepper\\_enable](#) ([StepperX](#) \*stepperx)  
*Enables the stepper motor driver.*
- void [Stepper\\_disable](#) ([StepperX](#) \*stepperx)  
*Disables the stepper motor driver.*
- void [Stepper\\_setspeed](#) ([StepperX](#) \*stepperx, uint16\_t speed, uint8\_t dir)  
*Sets the speed and direction of the stepper motor.*
- void [SysTick\\_Init](#) (void)  
*Initializes the system tick for delay functions.*
- void [Delay\\_us](#) (uint32\_t us)  
*Delays the program execution for a specified number of microseconds.*

### 9.13.1 Detailed Description

Header for stepper\_driver.c file. This file contains the common defines of the application.

Created on: May 17, 2024

Author: vvinh

### 9.13.2 Function Documentation

#### 9.13.2.1 Delay\_us()

```
void Delay_us (
    uint32_t us )
```

Delays the program execution for a specified number of microseconds.

##### Parameters

<i>us</i>	Number of microseconds to delay
-----------	---------------------------------

#### 9.13.2.2 Stepper\_disable()

```
void Stepper_disable (
    StepperX * stepperx )
```

Disables the stepper motor driver.

##### Parameters

<i>stepperx</i>	Pointer to the <a href="#">StepperX</a> structure
-----------------	---

### 9.13.2.3 Stepper\_enable()

```
void Stepper_enable (
    StepperX * stepperx )
```

Enables the stepper motor driver.

#### Parameters

<i>stepperx</i>	Pointer to the <a href="#">StepperX</a> structure
-----------------	---

### 9.13.2.4 Stepper\_init()

```
void Stepper_init (
    StepperX * stepperx,
    GPIO_TypeDef * GPIO,
    uint16_t EN_PIN,
    uint16_t DIR_PIN,
    uint16_t STP_PIN )
```

Initializes the stepper motor driver with the specified GPIO port and pins.

#### Parameters

<i>stepperx</i>	Pointer to the <a href="#">StepperX</a> structure
<i>GPIO</i>	GPIO port
<i>EN_PIN</i>	Enable pin
<i>DIR_PIN</i>	Direction pin
<i>STP_PIN</i>	Step pin

### 9.13.2.5 Stepper\_setspeed()

```
void Stepper_setspeed (
    StepperX * stepperx,
    uint16_t speed,
    uint8_t dir )
```

Sets the speed and direction of the stepper motor.

#### Parameters

<i>stepperx</i>	Pointer to the <a href="#">StepperX</a> structure
<i>speed</i>	Speed of the stepper motor
<i>dir</i>	Direction of the stepper motor (1 for one direction, 0 for the opposite)

### 9.13.2.6 SysTick\_Init()

```
void SysTick_Init (
```

```
void )
```

Initializes the system tick for delay functions.

## 9.14 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/switch.h File Reference

Header for switch.c file. This file contains the common defines of the application.

```
#include <stdio.h>
#include <stdint.h>
#include "stm32f4xx_hal.h"
```

### Data Structures

- struct [SwitchX](#)  
*Structure to define a switch with GPIO port, pin, and status.*

### Functions

- void [Switch\\_init](#) ([SwitchX](#) \*switchx, GPIO\_TypeDef \*GPIO, uint16\_t PIN)  
*Initializes the switch with the specified GPIO port and pin.*
- uint8\_t [Switch\\_getStatus](#) ([SwitchX](#) \*switchx)  
*Gets the status of the switch.*

#### 9.14.1 Detailed Description

Header for switch.c file. This file contains the common defines of the application.

Created on: May 17, 2024

Author: vvinh

#### 9.14.2 Function Documentation

##### 9.14.2.1 Switch\_getStatus()

```
uint8_t Switch_getStatus (
    SwitchX * switchx )
```

Gets the status of the switch.

#### Parameters

<i>switchx</i>	Pointer to the <a href="#">SwitchX</a> structure
----------------	--

**Returns**

uint8\_t Status of the switch (1 if pressed, 0 if not pressed)

**9.14.2.2 Switch\_init()**

```
void Switch_init (
    SwitchX * switchx,
    GPIO_TypeDef * GPIO,
    uint16_t PIN )
```

Initializes the switch with the specified GPIO port and pin.

**Parameters**

<i>switchx</i>	Pointer to the <a href="#">SwitchX</a> structure
<i>GPIO</i>	GPIO port
<i>PIN</i>	GPIO pin

## 9.15 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/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.*

**9.15.1 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.

**9.15.2 Function Documentation****9.15.2.1 Error\_Handler()**

```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

Return values

None	
------	--

## 9.16 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/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 [HAL\\_TIM\\_MspPostInit](#) (TIM\_HandleTypeDef \*htim)
- void [Error\\_Handler](#) (void)

*This function is executed in case of error occurrence.*

### 9.16.1 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.

### 9.16.2 Function Documentation

#### 9.16.2.1 Error\_Handler()

```
void Error_Handler (  
    void )
```

This function is executed in case of error occurrence.

Return values

None	
------	--



### 9.16.2.2 HAL\_TIM\_MspPostInit()

```
void HAL_TIM_MspPostInit (
    TIM_HandleTypeDef * htim )
```

## 9.17 Ball\_Launcher\_Controller/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.*

### 9.17.1 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.

### 9.17.2 Function Documentation

#### 9.17.2.1 Error\_Handler()

```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

#### Return values

None	
------	--

## 9.18 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/↵ Inc/mpu6050.h File Reference

Header for mpu6050.c file. This file contains the common defines of the application.

```
#include <stdio.h>
#include <stdint.h>
#include "stm32f4xx_hal.h"
```

### Data Structures

- struct [MPU6050](#)  
*Struct representing a mpu6050 imu sensor.*

### Functions

- uint16\_t [mpu6050\\_init](#) ([MPU6050](#) \*imux, I2C\_HandleTypeDef \*hi2c)  
*Initializes the [MPU6050](#) sensor.*
- void [mpu6050\\_calibrate](#) ([MPU6050](#) \*imux)  
*Calibrates the [MPU6050](#) sensor.*
- void [mpu6050\\_update](#) ([MPU6050](#) \*imux)  
*Updates the [MPU6050](#) sensor data.*
- int16\_t [mpu6050\\_get\\_gX](#) ([MPU6050](#) \*imux)  
*Gets the calibrated X-axis gyroscope data.*
- int16\_t [mpu6050\\_get\\_gY](#) ([MPU6050](#) \*imux)  
*Gets the calibrated Y-axis gyroscope data.*
- int16\_t [mpu6050\\_get\\_gZ](#) ([MPU6050](#) \*imux)  
*Gets the calibrated Z-axis gyroscope data.*
- int32\_t [mpu6050\\_get\\_X](#) ([MPU6050](#) \*imux)  
*Gets the X-axis accelerometer data.*
- int32\_t [mpu6050\\_get\\_Y](#) ([MPU6050](#) \*imux)  
*Gets the Y-axis accelerometer data.*
- int32\_t [mpu6050\\_get\\_Z](#) ([MPU6050](#) \*imux)  
*Gets the Z-axis accelerometer data.*

### 9.18.1 Detailed Description

Header for mpu6050.c file. This file contains the common defines of the application.

### 9.18.2 Function Documentation

#### 9.18.2.1 mpu6050\_calibrate()

```
void mpu6050_calibrate (
    MPU6050 * imux )
```

Calibrates the [MPU6050](#) sensor.

#### Parameters

<i>imux</i>	Pointer to the <a href="#">MPU6050</a> structure.
-------------	---

#### 9.18.2.2 mpu6050\_get\_gX()

```
int16_t mpu6050_get_gX (  
    MPU6050 * imux )
```

Gets the calibrated X-axis gyroscope data.

#### Parameters

<i>imux</i>	Pointer to the <a href="#">MPU6050</a> structure.
-------------	---

#### Returns

int16\_t Calibrated X-axis gyroscope data.

#### 9.18.2.3 mpu6050\_get\_gY()

```
int16_t mpu6050_get_gY (  
    MPU6050 * imux )
```

Gets the calibrated Y-axis gyroscope data.

#### Parameters

<i>imux</i>	Pointer to the <a href="#">MPU6050</a> structure.
-------------	---

#### Returns

int16\_t Calibrated Y-axis gyroscope data.

#### 9.18.2.4 mpu6050\_get\_gZ()

```
int16_t mpu6050_get_gZ (  
    MPU6050 * imux )
```

Gets the calibrated Z-axis gyroscope data.

#### Parameters

<i>imux</i>	Pointer to the <a href="#">MPU6050</a> structure.
-------------	---

**Returns**

int16\_t Calibrated Z-axis gyroscope data.

**9.18.2.5 mpu6050\_get\_X()**

```
int32_t mpu6050_get_X (
    MPU6050 * imux )
```

Gets the X-axis accelerometer data.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
-------------	-----------------------------------

**Returns**

int32\_t X-axis accelerometer data.

**9.18.2.6 mpu6050\_get\_Y()**

```
int32_t mpu6050_get_Y (
    MPU6050 * imux )
```

Gets the Y-axis accelerometer data.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
-------------	-----------------------------------

**Returns**

int32\_t Y-axis accelerometer data.

**9.18.2.7 mpu6050\_get\_Z()**

```
int32_t mpu6050_get_Z (
    MPU6050 * imux )
```

Gets the Z-axis accelerometer data.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
-------------	-----------------------------------

**Returns**

int32\_t Z-axis accelerometer data.

**9.18.2.8 mpu6050\_init()**

```
uint16_t mpu6050_init (
    MPU6050 * imux,
    I2C_HandleTypeDef * hi2c )
```

Initializes the MPU6050 sensor.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
<i>hi2c</i>	Pointer to the I2C handle structure.

**Returns**

uint16\_t Returns 1 if initialization is successful, otherwise 0.

**9.18.2.9 mpu6050\_update()**

```
void mpu6050_update (
    MPU6050 * imux )
```

Updates the MPU6050 sensor data.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
<i>dt</i>	Time interval in milliseconds.

**9.19 Ball\_Launcher\_Controller/Core/Inc/mpu6050.h File Reference**

Header for mpu6050.c file. This file contains the common defines of the application.

```
#include <stdio.h>
#include <stdint.h>
#include "stm32f4xx_hal.h"
```

**Data Structures**

- struct MPU6050

*Struct representing a mpu6050 imu sensor.*

## Functions

- `uint16_t mpu6050_init (MPU6050 *imux, I2C_HandleTypeDef *hi2c)`  
*Initializes the MPU6050 sensor.*
- `void mpu6050_calibrate (MPU6050 *imux)`  
*Calibrates the MPU6050 sensor.*
- `void mpu6050_update (MPU6050 *imux)`  
*Updates the MPU6050 sensor data.*
- `int16_t mpu6050_get_gX (MPU6050 *imux)`  
*Gets the calibrated X-axis gyroscope data.*
- `int16_t mpu6050_get_gY (MPU6050 *imux)`  
*Gets the calibrated Y-axis gyroscope data.*
- `int16_t mpu6050_get_gZ (MPU6050 *imux)`  
*Gets the calibrated Z-axis gyroscope data.*
- `int32_t mpu6050_get_X (MPU6050 *imux)`  
*Gets the X-axis accelerometer data.*
- `int32_t mpu6050_get_Y (MPU6050 *imux)`  
*Gets the Y-axis accelerometer data.*
- `int32_t mpu6050_get_Z (MPU6050 *imux)`  
*Gets the Z-axis accelerometer data.*

### 9.19.1 Detailed Description

Header for mpu6050.c file. This file contains the common defines of the application.

### 9.19.2 Function Documentation

#### 9.19.2.1 mpu6050\_calibrate()

```
void mpu6050_calibrate (
    MPU6050 * imux )
```

Calibrates the MPU6050 sensor.

##### Parameters

<i>imux</i>	Pointer to the MPU6050 structure.
-------------	-----------------------------------

#### 9.19.2.2 mpu6050\_get\_gX()

```
int16_t mpu6050_get_gX (
    MPU6050 * imux )
```

Gets the calibrated X-axis gyroscope data.

##### Parameters

<i>imux</i>	Pointer to the MPU6050 structure.
-------------	-----------------------------------

**Returns**

int16\_t Calibrated X-axis gyroscope data.

**9.19.2.3 mpu6050\_get\_gY()**

```
int16_t mpu6050_get_gY (  
    MPU6050 * imux )
```

Gets the calibrated Y-axis gyroscope data.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
-------------	-----------------------------------

**Returns**

int16\_t Calibrated Y-axis gyroscope data.

**9.19.2.4 mpu6050\_get\_gZ()**

```
int16_t mpu6050_get_gZ (  
    MPU6050 * imux )
```

Gets the calibrated Z-axis gyroscope data.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
-------------	-----------------------------------

**Returns**

int16\_t Calibrated Z-axis gyroscope data.

**9.19.2.5 mpu6050\_get\_X()**

```
int32_t mpu6050_get_X (  
    MPU6050 * imux )
```

Gets the X-axis accelerometer data.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
-------------	-----------------------------------

**Returns**

int32\_t X-axis accelerometer data.

**9.19.2.6 mpu6050\_get\_Y()**

```
int32_t mpu6050_get_Y (
    MPU6050 * imux )
```

Gets the Y-axis accelerometer data.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
-------------	-----------------------------------

**Returns**

int32\_t Y-axis accelerometer data.

**9.19.2.7 mpu6050\_get\_Z()**

```
int32_t mpu6050_get_Z (
    MPU6050 * imux )
```

Gets the Z-axis accelerometer data.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
-------------	-----------------------------------

**Returns**

int32\_t Z-axis accelerometer data.

**9.19.2.8 mpu6050\_init()**

```
uint16_t mpu6050_init (
    MPU6050 * imux,
    I2C_HandleTypeDef * hi2c )
```

Initializes the MPU6050 sensor.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
<i>hi2c</i>	Pointer to the I2C handle structure.



**Returns**

uint16\_t Returns 1 if initialization is successful, otherwise 0.

**9.19.2.9 mpu6050\_update()**

```
void mpu6050_update (
    MPU6050 * imux )
```

Updates the MPU6050 sensor data.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
<i>dt</i>	Time interval in milliseconds.

## 9.20 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/↵ Inc/stm32f4xx\_hal\_conf.h File Reference

```
#include "stm32f4xx_hal_rcc.h"
#include "stm32f4xx_hal_gpio.h"
#include "stm32f4xx_hal_exti.h"
#include "stm32f4xx_hal_dma.h"
#include "stm32f4xx_hal_cortex.h"
#include "stm32f4xx_hal_flash.h"
#include "stm32f4xx_hal_i2c.h"
#include "stm32f4xx_hal_pwr.h"
#include "stm32f4xx_hal_uart.h"
```

**Macros**

- #define HAL\_MODULE\_ENABLED
- *This is the list of modules to be used in the HAL driver.*
- #define HAL\_I2C\_MODULE\_ENABLED
- #define HAL\_UART\_MODULE\_ENABLED
- #define HAL\_GPIO\_MODULE\_ENABLED
- #define HAL\_EXTI\_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 25000000U

*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 100U
- #define HSI\_VALUE ((uint32\_t)16000000U)

*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 32000U

*Internal Low Speed oscillator (LSI) value.*

- #define LSE\_VALUE 32768U

*External Low Speed oscillator (LSE) value.*

- #define LSE\_STARTUP\_TIMEOUT 5000U

- #define EXTERNAL\_CLOCK\_VALUE 12288000U

*External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S\_CKIN pad.*

- #define VDD\_VALUE 3300U

*This is the HAL system configuration section.*

- #define TICK\_INT\_PRIORITY 15U
- #define USE\_RTOS 0U
- #define PREFETCH\_ENABLE 1U
- #define INSTRUCTION\_CACHE\_ENABLE 1U
- #define DATA\_CACHE\_ENABLE 1U
- #define USE\_HAL\_ADC\_REGISTER\_CALLBACKS 0U /\* ADC register callback disabled \*/
- #define USE\_HAL\_CAN\_REGISTER\_CALLBACKS 0U /\* CAN register callback disabled \*/
- #define USE\_HAL\_CEC\_REGISTER\_CALLBACKS 0U /\* CEC register callback disabled \*/
- #define USE\_HAL\_CRYPT\_REGISTER\_CALLBACKS 0U /\* CRYPT register callback disabled \*/
- #define USE\_HAL\_DAC\_REGISTER\_CALLBACKS 0U /\* DAC register callback disabled \*/
- #define USE\_HAL\_DCMI\_REGISTER\_CALLBACKS 0U /\* DCMI register callback disabled \*/
- #define USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS 0U /\* DFSDM register callback disabled \*/
- #define USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS 0U /\* DMA2D register callback disabled \*/
- #define USE\_HAL\_DSI\_REGISTER\_CALLBACKS 0U /\* DSI register callback disabled \*/
- #define USE\_HAL\_ETH\_REGISTER\_CALLBACKS 0U /\* ETH register callback disabled \*/
- #define USE\_HAL\_HASH\_REGISTER\_CALLBACKS 0U /\* HASH register callback disabled \*/
- #define USE\_HAL\_HCD\_REGISTER\_CALLBACKS 0U /\* HCD register callback disabled \*/
- #define USE\_HAL\_I2C\_REGISTER\_CALLBACKS 0U /\* I2C register callback disabled \*/
- #define USE\_HAL\_FMPI2C\_REGISTER\_CALLBACKS 0U /\* FMPI2C register callback disabled \*/
- #define USE\_HAL\_FMPMBUS\_REGISTER\_CALLBACKS 0U /\* FMPMBUS register callback disabled \*/
- #define USE\_HAL\_I2S\_REGISTER\_CALLBACKS 0U /\* I2S register callback disabled \*/
- #define USE\_HAL\_IRDA\_REGISTER\_CALLBACKS 0U /\* IRDA register callback disabled \*/
- #define USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS 0U /\* LPTIM register callback disabled \*/
- #define USE\_HAL\_LTDC\_REGISTER\_CALLBACKS 0U /\* LTDC register callback disabled \*/
- #define USE\_HAL\_MMC\_REGISTER\_CALLBACKS 0U /\* MMC register callback disabled \*/
- #define USE\_HAL\_NAND\_REGISTER\_CALLBACKS 0U /\* NAND register callback disabled \*/
- #define USE\_HAL\_NOR\_REGISTER\_CALLBACKS 0U /\* NOR register callback disabled \*/
- #define USE\_HAL\_PCCARD\_REGISTER\_CALLBACKS 0U /\* PCCARD register callback disabled \*/
- #define USE\_HAL\_PCD\_REGISTER\_CALLBACKS 0U /\* PCD register callback disabled \*/
- #define USE\_HAL\_QSPI\_REGISTER\_CALLBACKS 0U /\* QSPI register callback disabled \*/
- #define USE\_HAL\_RNG\_REGISTER\_CALLBACKS 0U /\* RNG register callback disabled \*/
- #define USE\_HAL\_RTC\_REGISTER\_CALLBACKS 0U /\* RTC register callback disabled \*/
- #define USE\_HAL\_SAI\_REGISTER\_CALLBACKS 0U /\* SAI register callback disabled \*/
- #define USE\_HAL\_SD\_REGISTER\_CALLBACKS 0U /\* SD register callback disabled \*/
- #define USE\_HAL\_SMARTCARD\_REGISTER\_CALLBACKS 0U /\* SMARTCARD register callback disabled \*/
- #define USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS 0U /\* SDRAM register callback disabled \*/
- #define USE\_HAL\_SRAM\_REGISTER\_CALLBACKS 0U /\* SRAM register callback disabled \*/
- #define USE\_HAL\_SPDIFRX\_REGISTER\_CALLBACKS 0U /\* SPDIFRX register callback disabled \*/
- #define USE\_HAL\_SMBUS\_REGISTER\_CALLBACKS 0U /\* SMBUS register callback disabled \*/
- #define USE\_HAL\_SPI\_REGISTER\_CALLBACKS 0U /\* SPI register callback disabled \*/
- #define USE\_HAL\_TIM\_REGISTER\_CALLBACKS 0U /\* TIM register callback disabled \*/

- #define `USE_HAL_UART_REGISTER_CALLBACKS` 0U /\* UART register callback disabled \*/
- #define `USE_HAL_USART_REGISTER_CALLBACKS` 0U /\* USART register callback disabled \*/
- #define `USE_HAL_WWDG_REGISTER_CALLBACKS` 0U /\* WWDG register callback disabled \*/
- #define `MAC_ADDR0` 2U

*Uncomment the line below to expand the "assert\_param" macro in the HAL drivers code.*

- #define `MAC_ADDR1` 0U
- #define `MAC_ADDR2` 0U
- #define `MAC_ADDR3` 0U
- #define `MAC_ADDR4` 0U
- #define `MAC_ADDR5` 0U
- #define `ETH_RX_BUF_SIZE` ETH\_MAX\_PACKET\_SIZE /\* buffer size for receive \*/
- #define `ETH_TX_BUF_SIZE` ETH\_MAX\_PACKET\_SIZE /\* buffer size for transmit \*/
- #define `ETH_RXBUFNB` 4U /\* 4 Rx buffers of size ETH\_RX\_BUF\_SIZE \*/
- #define `ETH_TXBUFNB` 4U /\* 4 Tx buffers of size ETH\_TX\_BUF\_SIZE \*/
- #define `DP83848_PHY_ADDRESS`
- #define `PHY_RESET_DELAY` 0x000000FFU
- #define `PHY_CONFIG_DELAY` 0x00000FFFU
- #define `PHY_READ_TO` 0x0000FFFFU
- #define `PHY_WRITE_TO` 0x0000FFFFU
- #define `PHY_BCR` ((uint16\_t)0x0000U)
- #define `PHY_BSR` ((uint16\_t)0x0001U)
- #define `PHY_RESET` ((uint16\_t)0x8000U)
- #define `PHY_LOOPBACK` ((uint16\_t)0x4000U)
- #define `PHY_FULLDUPLEX_100M` ((uint16\_t)0x2100U)
- #define `PHY_HALFDUPLEX_100M` ((uint16\_t)0x2000U)
- #define `PHY_FULLDUPLEX_10M` ((uint16\_t)0x0100U)
- #define `PHY_HALFDUPLEX_10M` ((uint16\_t)0x0000U)
- #define `PHY_AUTONEGOTIATION` ((uint16\_t)0x1000U)
- #define `PHY_RESTART_AUTONEGOTIATION` ((uint16\_t)0x0200U)
- #define `PHY_POWERDOWN` ((uint16\_t)0x0800U)
- #define `PHY_ISOLATE` ((uint16\_t)0x0400U)
- #define `PHY_AUTONEGO_COMPLETE` ((uint16\_t)0x0020U)
- #define `PHY_LINKED_STATUS` ((uint16\_t)0x0004U)
- #define `PHY_JABBER_DETECTION` ((uint16\_t)0x0002U)
- #define `PHY_SR` ((uint16\_t))
- #define `PHY_SPEED_STATUS` ((uint16\_t))
- #define `PHY_DUPLEX_STATUS` ((uint16\_t))
- #define `USE_SPI_CRC` 0U
- #define `assert_param`(expr) ((void)0U)

*Include module's header file.*

## 9.20.1 Macro Definition Documentation

### 9.20.1.1 assert\_param

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

Include module's header file.

### 9.20.1.2 DATA\_CACHE\_ENABLE

```
#define DATA_CACHE_ENABLE 1U
```

### 9.20.1.3 DP83848\_PHY\_ADDRESS

```
#define DP83848_PHY_ADDRESS
```

### 9.20.1.4 ETH\_RX\_BUF\_SIZE

```
#define ETH_RX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for receive */
```

### 9.20.1.5 ETH\_RXBUFNB

```
#define ETH_RXBUFNB 4U /* 4 Rx buffers of size ETH_RX_BUF_SIZE */
```

### 9.20.1.6 ETH\_TX\_BUF\_SIZE

```
#define ETH_TX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for transmit */
```

### 9.20.1.7 ETH\_TXBUFNB

```
#define ETH_TXBUFNB 4U /* 4 Tx buffers of size ETH_TX_BUF_SIZE */
```

### 9.20.1.8 EXTERNAL\_CLOCK\_VALUE

```
#define EXTERNAL_CLOCK_VALUE 12288000U
```

External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S\_CKIN pad.

Value of the External audio frequency in Hz

### 9.20.1.9 HAL\_CORTEX\_MODULE\_ENABLED

```
#define HAL_CORTEX_MODULE_ENABLED
```

### 9.20.1.10 HAL\_DMA\_MODULE\_ENABLED

```
#define HAL_DMA_MODULE_ENABLED
```

#### 9.20.1.11 HAL\_EXTI\_MODULE\_ENABLED

```
#define HAL_EXTI_MODULE_ENABLED
```

#### 9.20.1.12 HAL\_FLASH\_MODULE\_ENABLED

```
#define HAL_FLASH_MODULE_ENABLED
```

#### 9.20.1.13 HAL\_GPIO\_MODULE\_ENABLED

```
#define HAL_GPIO_MODULE_ENABLED
```

#### 9.20.1.14 HAL\_I2C\_MODULE\_ENABLED

```
#define HAL_I2C_MODULE_ENABLED
```

#### 9.20.1.15 HAL\_MODULE\_ENABLED

```
#define HAL_MODULE_ENABLED
```

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

#### 9.20.1.16 HAL\_PWR\_MODULE\_ENABLED

```
#define HAL_PWR_MODULE_ENABLED
```

#### 9.20.1.17 HAL\_RCC\_MODULE\_ENABLED

```
#define HAL_RCC_MODULE_ENABLED
```

#### 9.20.1.18 HAL\_UART\_MODULE\_ENABLED

```
#define HAL_UART_MODULE_ENABLED
```

#### 9.20.1.19 HSE\_STARTUP\_TIMEOUT

```
#define HSE_STARTUP_TIMEOUT 100U
```

Time out for HSE start up, in ms

### 9.20.1.20 HSE\_VALUE

```
#define HSE_VALUE 25000000U
```

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

### 9.20.1.21 HSI\_VALUE

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

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

### 9.20.1.22 INSTRUCTION\_CACHE\_ENABLE

```
#define INSTRUCTION_CACHE_ENABLE 1U
```

### 9.20.1.23 LSE\_STARTUP\_TIMEOUT

```
#define LSE_STARTUP_TIMEOUT 5000U
```

Time out for LSE start up, in ms

### 9.20.1.24 LSE\_VALUE

```
#define LSE_VALUE 32768U
```

External Low Speed oscillator (LSE) value.

< 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 Low Speed oscillator in Hz

### 9.20.1.25 LSI\_VALUE

```
#define LSI_VALUE 32000U
```

Internal Low Speed oscillator (LSI) value.

LSI Typical Value in Hz

#### 9.20.1.26 MAC\_ADDR0

```
#define MAC_ADDR0 2U
```

Uncomment the line below to expanse the "assert\_param" macro in the HAL drivers code.

#### 9.20.1.27 MAC\_ADDR1

```
#define MAC_ADDR1 0U
```

#### 9.20.1.28 MAC\_ADDR2

```
#define MAC_ADDR2 0U
```

#### 9.20.1.29 MAC\_ADDR3

```
#define MAC_ADDR3 0U
```

#### 9.20.1.30 MAC\_ADDR4

```
#define MAC_ADDR4 0U
```

#### 9.20.1.31 MAC\_ADDR5

```
#define MAC_ADDR5 0U
```

#### 9.20.1.32 PHY\_AUTONEGO\_COMPLETE

```
#define PHY_AUTONEGO_COMPLETE ((uint16_t)0x0020U)
```

Auto-Negotiation process completed

#### 9.20.1.33 PHY\_AUTONEGOTIATION

```
#define PHY_AUTONEGOTIATION ((uint16_t)0x1000U)
```

Enable auto-negotiation function

#### 9.20.1.34 PHY\_BCR

```
#define PHY_BCR ((uint16_t)0x0000U)
```

Transceiver Basic Control Register

#### 9.20.1.35 PHY\_BSR

```
#define PHY_BSR ((uint16_t)0x0001U)
```

Transceiver Basic Status Register

#### 9.20.1.36 PHY\_CONFIG\_DELAY

```
#define PHY_CONFIG_DELAY 0x00000FFFU
```

#### 9.20.1.37 PHY\_DUPLEX\_STATUS

```
#define PHY_DUPLEX_STATUS ((uint16_t))
```

PHY Duplex mask

#### 9.20.1.38 PHY\_FULLDUPLEX\_100M

```
#define PHY_FULLDUPLEX_100M ((uint16_t)0x2100U)
```

Set the full-duplex mode at 100 Mb/s

#### 9.20.1.39 PHY\_FULLDUPLEX\_10M

```
#define PHY_FULLDUPLEX_10M ((uint16_t)0x0100U)
```

Set the full-duplex mode at 10 Mb/s

#### 9.20.1.40 PHY\_HALFDUPLEX\_100M

```
#define PHY_HALFDUPLEX_100M ((uint16_t)0x2000U)
```

Set the half-duplex mode at 100 Mb/s

#### 9.20.1.41 PHY\_HALFDUPLEX\_10M

```
#define PHY_HALFDUPLEX_10M ((uint16_t)0x0000U)
```

Set the half-duplex mode at 10 Mb/s



#### 9.20.1.42 PHY\_ISOLATE

```
#define PHY_ISOLATE ((uint16_t)0x0400U)
```

Isolate PHY from MII

#### 9.20.1.43 PHY\_JABBER\_DETECTION

```
#define PHY_JABBER_DETECTION ((uint16_t)0x0002U)
```

Jabber condition detected

#### 9.20.1.44 PHY\_LINKED\_STATUS

```
#define PHY_LINKED_STATUS ((uint16_t)0x0004U)
```

Valid link established

#### 9.20.1.45 PHY\_LOOPBACK

```
#define PHY_LOOPBACK ((uint16_t)0x4000U)
```

Select loop-back mode

#### 9.20.1.46 PHY\_POWERDOWN

```
#define PHY_POWERDOWN ((uint16_t)0x0800U)
```

Select the power down mode

#### 9.20.1.47 PHY\_READ\_TO

```
#define PHY_READ_TO 0x0000FFFFU
```

#### 9.20.1.48 PHY\_RESET

```
#define PHY_RESET ((uint16_t)0x8000U)
```

PHY Reset

#### 9.20.1.49 PHY\_RESET\_DELAY

```
#define PHY_RESET_DELAY 0x000000FFU
```

#### 9.20.1.50 PHY\_RESTART\_AUTONEGOTIATION

```
#define PHY_RESTART_AUTONEGOTIATION ((uint16_t)0x0200U)
```

Restart auto-negotiation function

#### 9.20.1.51 PHY\_SPEED\_STATUS

```
#define PHY_SPEED_STATUS ((uint16_t))
```

PHY Speed mask

#### 9.20.1.52 PHY\_SR

```
#define PHY_SR ((uint16_t))
```

PHY status register Offset

#### 9.20.1.53 PHY\_WRITE\_TO

```
#define PHY_WRITE_TO 0x0000FFFFU
```

#### 9.20.1.54 PREFETCH\_ENABLE

```
#define PREFETCH_ENABLE 1U
```

#### 9.20.1.55 TICK\_INT\_PRIORITY

```
#define TICK_INT_PRIORITY 15U
```

tick interrupt priority

#### 9.20.1.56 USE\_HAL\_ADC\_REGISTER\_CALLBACKS

```
#define USE_HAL_ADC_REGISTER_CALLBACKS 0U /* ADC register callback disabled */
```

#### **9.20.1.57 USE\_HAL\_CAN\_REGISTER\_CALLBACKS**

```
#define USE_HAL_CAN_REGISTER_CALLBACKS 0U /* CAN register callback disabled */
```

#### **9.20.1.58 USE\_HAL\_CEC\_REGISTER\_CALLBACKS**

```
#define USE_HAL_CEC_REGISTER_CALLBACKS 0U /* CEC register callback disabled */
```

#### **9.20.1.59 USE\_HAL\_CRYPT\_REGISTER\_CALLBACKS**

```
#define USE_HAL_CRYPT_REGISTER_CALLBACKS 0U /* CRYPT register callback disabled */
```

#### **9.20.1.60 USE\_HAL\_DAC\_REGISTER\_CALLBACKS**

```
#define USE_HAL_DAC_REGISTER_CALLBACKS 0U /* DAC register callback disabled */
```

#### **9.20.1.61 USE\_HAL\_DCMI\_REGISTER\_CALLBACKS**

```
#define USE_HAL_DCMI_REGISTER_CALLBACKS 0U /* DCMI register callback disabled */
```

#### **9.20.1.62 USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS**

```
#define USE_HAL_DFSDM_REGISTER_CALLBACKS 0U /* DFSDM register callback disabled */
```

#### **9.20.1.63 USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS**

```
#define USE_HAL_DMA2D_REGISTER_CALLBACKS 0U /* DMA2D register callback disabled */
```

#### **9.20.1.64 USE\_HAL\_DSI\_REGISTER\_CALLBACKS**

```
#define USE_HAL_DSI_REGISTER_CALLBACKS 0U /* DSI register callback disabled */
```

#### **9.20.1.65 USE\_HAL\_ETH\_REGISTER\_CALLBACKS**

```
#define USE_HAL_ETH_REGISTER_CALLBACKS 0U /* ETH register callback disabled */
```

#### **9.20.1.66 USE\_HAL\_FMPI2C\_REGISTER\_CALLBACKS**

```
#define USE_HAL_FMPI2C_REGISTER_CALLBACKS 0U /* FMPI2C register callback disabled */
```

#### 9.20.1.67 USE\_HAL\_FMPSMBUS\_REGISTER\_CALLBACKS

```
#define USE_HAL_FMPSMBUS_REGISTER_CALLBACKS 0U /* FMPSMBUS register callback disabled */
```

#### 9.20.1.68 USE\_HAL\_HASH\_REGISTER\_CALLBACKS

```
#define USE_HAL_HASH_REGISTER_CALLBACKS 0U /* HASH register callback disabled */
```

#### 9.20.1.69 USE\_HAL\_HCD\_REGISTER\_CALLBACKS

```
#define USE_HAL_HCD_REGISTER_CALLBACKS 0U /* HCD register callback disabled */
```

#### 9.20.1.70 USE\_HAL\_I2C\_REGISTER\_CALLBACKS

```
#define USE_HAL_I2C_REGISTER_CALLBACKS 0U /* I2C register callback disabled */
```

#### 9.20.1.71 USE\_HAL\_I2S\_REGISTER\_CALLBACKS

```
#define USE_HAL_I2S_REGISTER_CALLBACKS 0U /* I2S register callback disabled */
```

#### 9.20.1.72 USE\_HAL\_IRDA\_REGISTER\_CALLBACKS

```
#define USE_HAL_IRDA_REGISTER_CALLBACKS 0U /* IRDA register callback disabled */
```

#### 9.20.1.73 USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS

```
#define USE_HAL_LPTIM_REGISTER_CALLBACKS 0U /* LPTIM register callback disabled */
```

#### 9.20.1.74 USE\_HAL\_LTDC\_REGISTER\_CALLBACKS

```
#define USE_HAL_LTDC_REGISTER_CALLBACKS 0U /* LTDC register callback disabled */
```

#### 9.20.1.75 USE\_HAL\_MMC\_REGISTER\_CALLBACKS

```
#define USE_HAL_MMC_REGISTER_CALLBACKS 0U /* MMC register callback disabled */
```

#### 9.20.1.76 USE\_HAL\_NAND\_REGISTER\_CALLBACKS

```
#define USE_HAL_NAND_REGISTER_CALLBACKS 0U /* NAND register callback disabled */
```

#### **9.20.1.77 USE\_HAL\_NOR\_REGISTER\_CALLBACKS**

```
#define USE_HAL_NOR_REGISTER_CALLBACKS 0U /* NOR register callback disabled */
```

#### **9.20.1.78 USE\_HAL\_PCCARD\_REGISTER\_CALLBACKS**

```
#define USE_HAL_PCCARD_REGISTER_CALLBACKS 0U /* PCCARD register callback disabled */
```

#### **9.20.1.79 USE\_HAL\_PCD\_REGISTER\_CALLBACKS**

```
#define USE_HAL_PCD_REGISTER_CALLBACKS 0U /* PCD register callback disabled */
```

#### **9.20.1.80 USE\_HAL\_QSPI\_REGISTER\_CALLBACKS**

```
#define USE_HAL_QSPI_REGISTER_CALLBACKS 0U /* QSPI register callback disabled */
```

#### **9.20.1.81 USE\_HAL\_RNG\_REGISTER\_CALLBACKS**

```
#define USE_HAL_RNG_REGISTER_CALLBACKS 0U /* RNG register callback disabled */
```

#### **9.20.1.82 USE\_HAL\_RTC\_REGISTER\_CALLBACKS**

```
#define USE_HAL_RTC_REGISTER_CALLBACKS 0U /* RTC register callback disabled */
```

#### **9.20.1.83 USE\_HAL\_SAI\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SAI_REGISTER_CALLBACKS 0U /* SAI register callback disabled */
```

#### **9.20.1.84 USE\_HAL\_SD\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SD_REGISTER_CALLBACKS 0U /* SD register callback disabled */
```

#### **9.20.1.85 USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SDRAM_REGISTER_CALLBACKS 0U /* SDRAM register callback disabled */
```

#### **9.20.1.86 USE\_HAL\_SMARTCARD\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SMARTCARD_REGISTER_CALLBACKS 0U /* SMARTCARD register callback disabled */
```

#### 9.20.1.87 USE\_HAL\_SMBUS\_REGISTER\_CALLBACKS

```
#define USE_HAL_SMBUS_REGISTER_CALLBACKS 0U /* SMBUS register callback disabled */
```

#### 9.20.1.88 USE\_HAL\_SPDIFRX\_REGISTER\_CALLBACKS

```
#define USE_HAL_SPDIFRX_REGISTER_CALLBACKS 0U /* SPDIFRX register callback disabled */
```

#### 9.20.1.89 USE\_HAL\_SPI\_REGISTER\_CALLBACKS

```
#define USE_HAL_SPI_REGISTER_CALLBACKS 0U /* SPI register callback disabled */
```

#### 9.20.1.90 USE\_HAL\_SRAM\_REGISTER\_CALLBACKS

```
#define USE_HAL_SRAM_REGISTER_CALLBACKS 0U /* SRAM register callback disabled */
```

#### 9.20.1.91 USE\_HAL\_TIM\_REGISTER\_CALLBACKS

```
#define USE_HAL_TIM_REGISTER_CALLBACKS 0U /* TIM register callback disabled */
```

#### 9.20.1.92 USE\_HAL\_UART\_REGISTER\_CALLBACKS

```
#define USE_HAL_UART_REGISTER_CALLBACKS 0U /* UART register callback disabled */
```

#### 9.20.1.93 USE\_HAL\_USART\_REGISTER\_CALLBACKS

```
#define USE_HAL_USART_REGISTER_CALLBACKS 0U /* USART register callback disabled */
```

#### 9.20.1.94 USE\_HAL\_WWDG\_REGISTER\_CALLBACKS

```
#define USE_HAL_WWDG_REGISTER_CALLBACKS 0U /* WWDG register callback disabled */
```

#### 9.20.1.95 USE\_RTOS

```
#define USE_RTOS 0U
```

#### 9.20.1.96 USE\_SPI\_CRC

```
#define USE_SPI_CRC 0U
```

### 9.20.1.97 VDD\_VALUE

```
#define VDD_VALUE 3300U
```

This is the HAL system configuration section.

Value of VDD in mv

## 9.21 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/stm32f4xx\_hal\_conf.h File Reference

```
#include "stm32f4xx_hal_rcc.h"
#include "stm32f4xx_hal_gpio.h"
#include "stm32f4xx_hal_exti.h"
#include "stm32f4xx_hal_dma.h"
#include "stm32f4xx_hal_cortex.h"
#include "stm32f4xx_hal_flash.h"
#include "stm32f4xx_hal_pwr.h"
#include "stm32f4xx_hal_tim.h"
#include "stm32f4xx_hal_uart.h"
```

### Macros

- #define [HAL\\_MODULE\\_ENABLED](#)  
*This is the list of modules to be used in the HAL driver.*
- #define [HAL\\_TIM\\_MODULE\\_ENABLED](#)
- #define [HAL\\_UART\\_MODULE\\_ENABLED](#)
- #define [HAL\\_GPIO\\_MODULE\\_ENABLED](#)
- #define [HAL\\_EXTI\\_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](#) 25000000U  
*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](#) 100U
- #define [HSI\\_VALUE](#) ((uint32\_t)16000000U)  
*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](#) 32000U  
*Internal Low Speed oscillator (LSI) value.*
- #define [LSE\\_VALUE](#) 32768U  
*External Low Speed oscillator (LSE) value.*
- #define [LSE\\_STARTUP\\_TIMEOUT](#) 5000U
- #define [EXTERNAL\\_CLOCK\\_VALUE](#) 12288000U  
*External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S\_CKIN pad.*
- #define [VDD\\_VALUE](#) 3300U

*This is the HAL system configuration section.*

```

• #define TICK_INT_PRIORITY 15U
• #define USE_RTOS 0U
• #define PREFETCH_ENABLE 1U
• #define INSTRUCTION_CACHE_ENABLE 1U
• #define DATA_CACHE_ENABLE 1U
• #define USE_HAL_ADC_REGISTER_CALLBACKS 0U /* ADC register callback disabled */
• #define USE_HAL_CAN_REGISTER_CALLBACKS 0U /* CAN register callback disabled */
• #define USE_HAL_CEC_REGISTER_CALLBACKS 0U /* CEC register callback disabled */
• #define USE_HAL_Cryp_REGISTER_CALLBACKS 0U /* CRYP register callback disabled */
• #define USE_HAL_DAC_REGISTER_CALLBACKS 0U /* DAC register callback disabled */
• #define USE_HAL_DCMI_REGISTER_CALLBACKS 0U /* DCMI register callback disabled */
• #define USE_HAL_DFSDM_REGISTER_CALLBACKS 0U /* DFSDM register callback disabled */
• #define USE_HAL_DMA2D_REGISTER_CALLBACKS 0U /* DMA2D register callback disabled */
• #define USE_HAL_DSI_REGISTER_CALLBACKS 0U /* DSI register callback disabled */
• #define USE_HAL_ETH_REGISTER_CALLBACKS 0U /* ETH register callback disabled */
• #define USE_HAL_HASH_REGISTER_CALLBACKS 0U /* HASH register callback disabled */
• #define USE_HAL_HCD_REGISTER_CALLBACKS 0U /* HCD register callback disabled */
• #define USE_HAL_I2C_REGISTER_CALLBACKS 0U /* I2C register callback disabled */
• #define USE_HAL_FMPI2C_REGISTER_CALLBACKS 0U /* FMPI2C register callback disabled */
• #define USE_HAL_FMPMBUS_REGISTER_CALLBACKS 0U /* FMPSMBUS register callback disabled */
• #define USE_HAL_I2S_REGISTER_CALLBACKS 0U /* I2S register callback disabled */
• #define USE_HAL_IRDA_REGISTER_CALLBACKS 0U /* IRDA register callback disabled */
• #define USE_HAL_LPTIM_REGISTER_CALLBACKS 0U /* LPTIM register callback disabled */
• #define USE_HAL_LTDC_REGISTER_CALLBACKS 0U /* LTDC register callback disabled */
• #define USE_HAL_MMC_REGISTER_CALLBACKS 0U /* MMC register callback disabled */
• #define USE_HAL_NAND_REGISTER_CALLBACKS 0U /* NAND register callback disabled */
• #define USE_HAL_NOR_REGISTER_CALLBACKS 0U /* NOR register callback disabled */
• #define USE_HAL_PCCARD_REGISTER_CALLBACKS 0U /* PCCARD register callback disabled */
• #define USE_HAL_PCD_REGISTER_CALLBACKS 0U /* PCD register callback disabled */
• #define USE_HAL_QSPI_REGISTER_CALLBACKS 0U /* QSPI register callback disabled */
• #define USE_HAL_RNG_REGISTER_CALLBACKS 0U /* RNG register callback disabled */
• #define USE_HAL_RTC_REGISTER_CALLBACKS 0U /* RTC register callback disabled */
• #define USE_HAL_SAI_REGISTER_CALLBACKS 0U /* SAI register callback disabled */
• #define USE_HAL_SD_REGISTER_CALLBACKS 0U /* SD register callback disabled */
• #define USE_HAL_SMARTCARD_REGISTER_CALLBACKS 0U /* SMARTCARD register callback disabled
*/
• #define USE_HAL_SDRAM_REGISTER_CALLBACKS 0U /* SDRAM register callback disabled */
• #define USE_HAL_SRAM_REGISTER_CALLBACKS 0U /* SRAM register callback disabled */
• #define USE_HAL_SPDIFRX_REGISTER_CALLBACKS 0U /* SPDIFRX register callback disabled */
• #define USE_HAL_SMBUS_REGISTER_CALLBACKS 0U /* SMBUS register callback disabled */
• #define USE_HAL_SPI_REGISTER_CALLBACKS 0U /* SPI register callback disabled */
• #define USE_HAL_TIM_REGISTER_CALLBACKS 0U /* TIM register callback disabled */
• #define USE_HAL_UART_REGISTER_CALLBACKS 0U /* UART register callback disabled */
• #define USE_HAL_USART_REGISTER_CALLBACKS 0U /* USART register callback disabled */
• #define USE_HAL_WWDG_REGISTER_CALLBACKS 0U /* WWDG register callback disabled */
• #define MAC_ADDR0 2U

```

*Uncomment the line below to expanse the "assert\_param" macro in the HAL drivers code.*

```

• #define MAC_ADDR1 0U
• #define MAC_ADDR2 0U
• #define MAC_ADDR3 0U
• #define MAC_ADDR4 0U
• #define MAC_ADDR5 0U
• #define ETH_RX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for receive */

```



- #define `ETH_TX_BUF_SIZE` `ETH_MAX_PACKET_SIZE` /\* `buffer` size for transmit \*/
- #define `ETH_RXBUFNB` `4U` /\* 4 Rx buffers of size `ETH_RX_BUF_SIZE` \*/
- #define `ETH_TXBUFNB` `4U` /\* 4 Tx buffers of size `ETH_TX_BUF_SIZE` \*/
- #define `DP83848_PHY_ADDRESS`
- #define `PHY_RESET_DELAY` `0x000000FFU`
- #define `PHY_CONFIG_DELAY` `0x00000FFFU`
- #define `PHY_READ_TO` `0x0000FFFFU`
- #define `PHY_WRITE_TO` `0x0000FFFFU`
- #define `PHY_BCR` `((uint16_t)0x0000U)`
- #define `PHY_BSR` `((uint16_t)0x0001U)`
- #define `PHY_RESET` `((uint16_t)0x8000U)`
- #define `PHY_LOOPBACK` `((uint16_t)0x4000U)`
- #define `PHY_FULLDUPLEX_100M` `((uint16_t)0x2100U)`
- #define `PHY_HALFDUPLEX_100M` `((uint16_t)0x2000U)`
- #define `PHY_FULLDUPLEX_10M` `((uint16_t)0x0100U)`
- #define `PHY_HALFDUPLEX_10M` `((uint16_t)0x0000U)`
- #define `PHY_AUTONEGOTIATION` `((uint16_t)0x1000U)`
- #define `PHY_RESTART_AUTONEGOTIATION` `((uint16_t)0x0200U)`
- #define `PHY_POWERDOWN` `((uint16_t)0x0800U)`
- #define `PHY_ISOLATE` `((uint16_t)0x0400U)`
- #define `PHY_AUTONEGO_COMPLETE` `((uint16_t)0x0020U)`
- #define `PHY_LINKED_STATUS` `((uint16_t)0x0004U)`
- #define `PHY_JABBER_DETECTION` `((uint16_t)0x0002U)`
- #define `PHY_SR` `((uint16_t))`
- #define `PHY_SPEED_STATUS` `((uint16_t))`
- #define `PHY_DUPLEX_STATUS` `((uint16_t))`
- #define `USE_SPI_CRC` `0U`
- #define `assert_param`(`expr`) `((void)0U)`

*Include module's header file.*

## 9.21.1 Macro Definition Documentation

### 9.21.1.1 `assert_param`

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

Include module's header file.

### 9.21.1.2 `DATA_CACHE_ENABLE`

```
#define DATA_CACHE_ENABLE 1U
```

### 9.21.1.3 `DP83848_PHY_ADDRESS`

```
#define DP83848_PHY_ADDRESS
```

### 9.21.1.4 `ETH_RX_BUF_SIZE`

```
#define ETH_RX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for receive */
```

#### 9.21.1.5 ETH\_RXBUFNB

```
#define ETH_RXBUFNB 4U /* 4 Rx buffers of size ETH_RX_BUF_SIZE */
```

#### 9.21.1.6 ETH\_TX\_BUF\_SIZE

```
#define ETH_TX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for transmit */
```

#### 9.21.1.7 ETH\_TXBUFNB

```
#define ETH_TXBUFNB 4U /* 4 Tx buffers of size ETH_TX_BUF_SIZE */
```

#### 9.21.1.8 EXTERNAL\_CLOCK\_VALUE

```
#define EXTERNAL_CLOCK_VALUE 12288000U
```

External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S\_CKIN pad.

Value of the External audio frequency in Hz

#### 9.21.1.9 HAL\_CORTEX\_MODULE\_ENABLED

```
#define HAL_CORTEX_MODULE_ENABLED
```

#### 9.21.1.10 HAL\_DMA\_MODULE\_ENABLED

```
#define HAL_DMA_MODULE_ENABLED
```

#### 9.21.1.11 HAL\_EXTI\_MODULE\_ENABLED

```
#define HAL_EXTI_MODULE_ENABLED
```

#### 9.21.1.12 HAL\_FLASH\_MODULE\_ENABLED

```
#define HAL_FLASH_MODULE_ENABLED
```

#### 9.21.1.13 HAL\_GPIO\_MODULE\_ENABLED

```
#define HAL_GPIO_MODULE_ENABLED
```

#### 9.21.1.14 HAL\_MODULE\_ENABLED

```
#define HAL_MODULE_ENABLED
```

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

#### 9.21.1.15 HAL\_PWR\_MODULE\_ENABLED

```
#define HAL_PWR_MODULE_ENABLED
```

#### 9.21.1.16 HAL\_RCC\_MODULE\_ENABLED

```
#define HAL_RCC_MODULE_ENABLED
```

#### 9.21.1.17 HAL\_TIM\_MODULE\_ENABLED

```
#define HAL_TIM_MODULE_ENABLED
```

#### 9.21.1.18 HAL\_UART\_MODULE\_ENABLED

```
#define HAL_UART_MODULE_ENABLED
```

#### 9.21.1.19 HSE\_STARTUP\_TIMEOUT

```
#define HSE_STARTUP_TIMEOUT 100U
```

Time out for HSE start up, in ms

#### 9.21.1.20 HSE\_VALUE

```
#define HSE_VALUE 25000000U
```

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

#### 9.21.1.21 HSI\_VALUE

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

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

#### 9.21.1.22 INSTRUCTION\_CACHE\_ENABLE

```
#define INSTRUCTION_CACHE_ENABLE 1U
```

#### 9.21.1.23 LSE\_STARTUP\_TIMEOUT

```
#define LSE_STARTUP_TIMEOUT 5000U
```

Time out for LSE start up, in ms

#### 9.21.1.24 LSE\_VALUE

```
#define LSE_VALUE 32768U
```

External Low Speed oscillator (LSE) value.

< 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 Low Speed oscillator in Hz

#### 9.21.1.25 LSI\_VALUE

```
#define LSI_VALUE 32000U
```

Internal Low Speed oscillator (LSI) value.

LSI Typical Value in Hz

#### 9.21.1.26 MAC\_ADDR0

```
#define MAC_ADDR0 2U
```

Uncomment the line below to expanse the "assert\_param" macro in the HAL drivers code.

#### 9.21.1.27 MAC\_ADDR1

```
#define MAC_ADDR1 0U
```

#### 9.21.1.28 MAC\_ADDR2

```
#define MAC_ADDR2 0U
```

#### 9.21.1.29 MAC\_ADDR3

```
#define MAC_ADDR3 0U
```

#### 9.21.1.30 MAC\_ADDR4

```
#define MAC_ADDR4 0U
```

#### 9.21.1.31 MAC\_ADDR5

```
#define MAC_ADDR5 0U
```

#### 9.21.1.32 PHY\_AUTONEGO\_COMPLETE

```
#define PHY_AUTONEGO_COMPLETE ((uint16_t)0x0020U)
```

Auto-Negotiation process completed

#### 9.21.1.33 PHY\_AUTONEGOTIATION

```
#define PHY_AUTONEGOTIATION ((uint16_t)0x1000U)
```

Enable auto-negotiation function

#### 9.21.1.34 PHY\_BCR

```
#define PHY_BCR ((uint16_t)0x0000U)
```

Transceiver Basic Control Register

#### 9.21.1.35 PHY\_BSR

```
#define PHY_BSR ((uint16_t)0x0001U)
```

Transceiver Basic Status Register

#### 9.21.1.36 PHY\_CONFIG\_DELAY

```
#define PHY_CONFIG_DELAY 0x00000FFFU
```

#### 9.21.1.37 PHY\_DUPLEX\_STATUS

```
#define PHY_DUPLEX_STATUS ((uint16_t))
```

PHY Duplex mask

#### 9.21.1.38 PHY\_FULLDUPLEX\_100M

```
#define PHY_FULLDUPLEX_100M ((uint16_t)0x2100U)
```

Set the full-duplex mode at 100 Mb/s

#### 9.21.1.39 PHY\_FULLDUPLEX\_10M

```
#define PHY_FULLDUPLEX_10M ((uint16_t)0x0100U)
```

Set the full-duplex mode at 10 Mb/s

#### 9.21.1.40 PHY\_HALFDUPLEX\_100M

```
#define PHY_HALFDUPLEX_100M ((uint16_t)0x2000U)
```

Set the half-duplex mode at 100 Mb/s

#### 9.21.1.41 PHY\_HALFDUPLEX\_10M

```
#define PHY_HALFDUPLEX_10M ((uint16_t)0x0000U)
```

Set the half-duplex mode at 10 Mb/s

#### 9.21.1.42 PHY\_ISOLATE

```
#define PHY_ISOLATE ((uint16_t)0x0400U)
```

Isolate PHY from MII

#### 9.21.1.43 PHY\_JABBER\_DETECTION

```
#define PHY_JABBER_DETECTION ((uint16_t)0x0002U)
```

Jabber condition detected

#### 9.21.1.44 PHY\_LINKED\_STATUS

```
#define PHY_LINKED_STATUS ((uint16_t)0x0004U)
```

Valid link established

#### 9.21.1.45 PHY\_LOOPBACK

```
#define PHY_LOOPBACK ((uint16_t)0x4000U)
```

Select loop-back mode

#### 9.21.1.46 PHY\_POWERDOWN

```
#define PHY_POWERDOWN ((uint16_t)0x0800U)
```

Select the power down mode

#### 9.21.1.47 PHY\_READ\_TO

```
#define PHY_READ_TO 0x0000FFFFU
```

#### 9.21.1.48 PHY\_RESET

```
#define PHY_RESET ((uint16_t)0x8000U)
```

PHY Reset

#### 9.21.1.49 PHY\_RESET\_DELAY

```
#define PHY_RESET_DELAY 0x000000FFU
```

#### 9.21.1.50 PHY\_RESTART\_AUTONEGOTIATION

```
#define PHY_RESTART_AUTONEGOTIATION ((uint16_t)0x0200U)
```

Restart auto-negotiation function

#### 9.21.1.51 PHY\_SPEED\_STATUS

```
#define PHY_SPEED_STATUS ((uint16_t))
```

PHY Speed mask

#### 9.21.1.52 PHY\_SR

```
#define PHY_SR ((uint16_t))
```

PHY status register Offset

#### 9.21.1.53 PHY\_WRITE\_TO

```
#define PHY_WRITE_TO 0x0000FFFFU
```

#### 9.21.1.54 PREFETCH\_ENABLE

```
#define PREFETCH_ENABLE 1U
```

#### 9.21.1.55 TICK\_INT\_PRIORITY

```
#define TICK_INT_PRIORITY 15U
```

tick interrupt priority

#### 9.21.1.56 USE\_HAL\_ADC\_REGISTER\_CALLBACKS

```
#define USE_HAL_ADC_REGISTER_CALLBACKS 0U /* ADC register callback disabled */
```

#### 9.21.1.57 USE\_HAL\_CAN\_REGISTER\_CALLBACKS

```
#define USE_HAL_CAN_REGISTER_CALLBACKS 0U /* CAN register callback disabled */
```

#### 9.21.1.58 USE\_HAL\_CEC\_REGISTER\_CALLBACKS

```
#define USE_HAL_CEC_REGISTER_CALLBACKS 0U /* CEC register callback disabled */
```

#### 9.21.1.59 USE\_HAL\_Cryp\_REGISTER\_CALLBACKS

```
#define USE_HAL_Cryp_REGISTER_CALLBACKS 0U /* CRYp register callback disabled */
```

#### 9.21.1.60 USE\_HAL\_DAC\_REGISTER\_CALLBACKS

```
#define USE_HAL_DAC_REGISTER_CALLBACKS 0U /* DAC register callback disabled */
```

#### 9.21.1.61 USE\_HAL\_DCMI\_REGISTER\_CALLBACKS

```
#define USE_HAL_DCMI_REGISTER_CALLBACKS 0U /* DCMI register callback disabled */
```

#### 9.21.1.62 USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS

```
#define USE_HAL_DFSDM_REGISTER_CALLBACKS 0U /* DFSDM register callback disabled */
```



#### 9.21.1.63 USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS

```
#define USE_HAL_DMA2D_REGISTER_CALLBACKS 0U /* DMA2D register callback disabled */
```

#### 9.21.1.64 USE\_HAL\_DSI\_REGISTER\_CALLBACKS

```
#define USE_HAL_DSI_REGISTER_CALLBACKS 0U /* DSI register callback disabled */
```

#### 9.21.1.65 USE\_HAL\_ETH\_REGISTER\_CALLBACKS

```
#define USE_HAL_ETH_REGISTER_CALLBACKS 0U /* ETH register callback disabled */
```

#### 9.21.1.66 USE\_HAL\_FMPI2C\_REGISTER\_CALLBACKS

```
#define USE_HAL_FMPI2C_REGISTER_CALLBACKS 0U /* FMPI2C register callback disabled */
```

#### 9.21.1.67 USE\_HAL\_FMPMBUS\_REGISTER\_CALLBACKS

```
#define USE_HAL_FMPMBUS_REGISTER_CALLBACKS 0U /* FMPMBUS register callback disabled */
```

#### 9.21.1.68 USE\_HAL\_HASH\_REGISTER\_CALLBACKS

```
#define USE_HAL_HASH_REGISTER_CALLBACKS 0U /* HASH register callback disabled */
```

#### 9.21.1.69 USE\_HAL\_HCD\_REGISTER\_CALLBACKS

```
#define USE_HAL_HCD_REGISTER_CALLBACKS 0U /* HCD register callback disabled */
```

#### 9.21.1.70 USE\_HAL\_I2C\_REGISTER\_CALLBACKS

```
#define USE_HAL_I2C_REGISTER_CALLBACKS 0U /* I2C register callback disabled */
```

#### 9.21.1.71 USE\_HAL\_I2S\_REGISTER\_CALLBACKS

```
#define USE_HAL_I2S_REGISTER_CALLBACKS 0U /* I2S register callback disabled */
```

#### 9.21.1.72 USE\_HAL\_IRDA\_REGISTER\_CALLBACKS

```
#define USE_HAL_IRDA_REGISTER_CALLBACKS 0U /* IRDA register callback disabled */
```

#### 9.21.1.73 USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS

```
#define USE_HAL_LPTIM_REGISTER_CALLBACKS 0U /* LPTIM register callback disabled */
```

#### 9.21.1.74 USE\_HAL\_LTDC\_REGISTER\_CALLBACKS

```
#define USE_HAL_LTDC_REGISTER_CALLBACKS 0U /* LTDC register callback disabled */
```

#### 9.21.1.75 USE\_HAL\_MMC\_REGISTER\_CALLBACKS

```
#define USE_HAL_MMC_REGISTER_CALLBACKS 0U /* MMC register callback disabled */
```

#### 9.21.1.76 USE\_HAL\_NAND\_REGISTER\_CALLBACKS

```
#define USE_HAL_NAND_REGISTER_CALLBACKS 0U /* NAND register callback disabled */
```

#### 9.21.1.77 USE\_HAL\_NOR\_REGISTER\_CALLBACKS

```
#define USE_HAL_NOR_REGISTER_CALLBACKS 0U /* NOR register callback disabled */
```

#### 9.21.1.78 USE\_HAL\_PCCARD\_REGISTER\_CALLBACKS

```
#define USE_HAL_PCCARD_REGISTER_CALLBACKS 0U /* PCCARD register callback disabled */
```

#### 9.21.1.79 USE\_HAL\_PCD\_REGISTER\_CALLBACKS

```
#define USE_HAL_PCD_REGISTER_CALLBACKS 0U /* PCD register callback disabled */
```

#### 9.21.1.80 USE\_HAL\_QSPI\_REGISTER\_CALLBACKS

```
#define USE_HAL_QSPI_REGISTER_CALLBACKS 0U /* QSPI register callback disabled */
```

#### 9.21.1.81 USE\_HAL\_RNG\_REGISTER\_CALLBACKS

```
#define USE_HAL_RNG_REGISTER_CALLBACKS 0U /* RNG register callback disabled */
```

#### 9.21.1.82 USE\_HAL\_RTC\_REGISTER\_CALLBACKS

```
#define USE_HAL_RTC_REGISTER_CALLBACKS 0U /* RTC register callback disabled */
```

#### 9.21.1.83 USE\_HAL\_SAI\_REGISTER\_CALLBACKS

```
#define USE_HAL_SAI_REGISTER_CALLBACKS 0U /* SAI register callback disabled */
```

#### 9.21.1.84 USE\_HAL\_SD\_REGISTER\_CALLBACKS

```
#define USE_HAL_SD_REGISTER_CALLBACKS 0U /* SD register callback disabled */
```

#### 9.21.1.85 USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS

```
#define USE_HAL_SDRAM_REGISTER_CALLBACKS 0U /* SDRAM register callback disabled */
```

#### 9.21.1.86 USE\_HAL\_SMARTCARD\_REGISTER\_CALLBACKS

```
#define USE_HAL_SMARTCARD_REGISTER_CALLBACKS 0U /* SMARTCARD register callback disabled */
```

#### 9.21.1.87 USE\_HAL\_SMBUS\_REGISTER\_CALLBACKS

```
#define USE_HAL_SMBUS_REGISTER_CALLBACKS 0U /* SMBUS register callback disabled */
```

#### 9.21.1.88 USE\_HAL\_SPDIFRX\_REGISTER\_CALLBACKS

```
#define USE_HAL_SPDIFRX_REGISTER_CALLBACKS 0U /* SPDIFRX register callback disabled */
```

#### 9.21.1.89 USE\_HAL\_SPI\_REGISTER\_CALLBACKS

```
#define USE_HAL_SPI_REGISTER_CALLBACKS 0U /* SPI register callback disabled */
```

#### 9.21.1.90 USE\_HAL\_SRAM\_REGISTER\_CALLBACKS

```
#define USE_HAL_SRAM_REGISTER_CALLBACKS 0U /* SRAM register callback disabled */
```

#### 9.21.1.91 USE\_HAL\_TIM\_REGISTER\_CALLBACKS

```
#define USE_HAL_TIM_REGISTER_CALLBACKS 0U /* TIM register callback disabled */
```

#### 9.21.1.92 USE\_HAL\_UART\_REGISTER\_CALLBACKS

```
#define USE_HAL_UART_REGISTER_CALLBACKS 0U /* UART register callback disabled */
```

### 9.21.1.93 USE\_HAL\_USART\_REGISTER\_CALLBACKS

```
#define USE_HAL_USART_REGISTER_CALLBACKS 0U /* USART register callback disabled */
```

### 9.21.1.94 USE\_HAL\_WWDG\_REGISTER\_CALLBACKS

```
#define USE_HAL_WWDG_REGISTER_CALLBACKS 0U /* WWDG register callback disabled */
```

### 9.21.1.95 USE\_RTOS

```
#define USE_RTOS 0U
```

### 9.21.1.96 USE\_SPI\_CRC

```
#define USE_SPI_CRC 0U
```

### 9.21.1.97 VDD\_VALUE

```
#define VDD_VALUE 3300U
```

This is the HAL system configuration section.

Value of VDD in mv

## 9.22 Ball\_Launcher\_Controller/Core/Inc/stm32f4xx\_hal\_conf.h File Reference

```
#include "stm32f4xx_hal_rcc.h"
#include "stm32f4xx_hal_gpio.h"
#include "stm32f4xx_hal_exti.h"
#include "stm32f4xx_hal_dma.h"
#include "stm32f4xx_hal_cortex.h"
#include "stm32f4xx_hal_flash.h"
#include "stm32f4xx_hal_i2c.h"
#include "stm32f4xx_hal_pwr.h"
#include "stm32f4xx_hal_uart.h"
```

## Macros

- #define `HAL_MODULE_ENABLED`  
*This is the list of modules to be used in the HAL driver.*
- #define `HAL_I2C_MODULE_ENABLED`
- #define `HAL_UART_MODULE_ENABLED`
- #define `HAL_GPIO_MODULE_ENABLED`
- #define `HAL_EXTI_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` 25000000U  
*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` 100U
- #define `HSI_VALUE` ((uint32\_t)16000000U)  
*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` 32000U  
*Internal Low Speed oscillator (LSI) value.*
- #define `LSE_VALUE` 32768U  
*External Low Speed oscillator (LSE) value.*
- #define `LSE_STARTUP_TIMEOUT` 5000U
- #define `EXTERNAL_CLOCK_VALUE` 12288000U  
*External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S\_CKIN pad.*
- #define `VDD_VALUE` 3300U  
*This is the HAL system configuration section.*
- #define `TICK_INT_PRIORITY` 15U
- #define `USE_RTOS` 0U
- #define `PREFETCH_ENABLE` 1U
- #define `INSTRUCTION_CACHE_ENABLE` 1U
- #define `DATA_CACHE_ENABLE` 1U
- #define `USE_HAL_ADC_REGISTER_CALLBACKS` 0U /\* ADC register callback disabled \*/
- #define `USE_HAL_CAN_REGISTER_CALLBACKS` 0U /\* CAN register callback disabled \*/
- #define `USE_HAL_CEC_REGISTER_CALLBACKS` 0U /\* CEC register callback disabled \*/
- #define `USE_HAL_Cryp_REGISTER_CALLBACKS` 0U /\* CRYP register callback disabled \*/
- #define `USE_HAL_DAC_REGISTER_CALLBACKS` 0U /\* DAC register callback disabled \*/
- #define `USE_HAL_DCMI_REGISTER_CALLBACKS` 0U /\* DCMI register callback disabled \*/
- #define `USE_HAL_DFSDM_REGISTER_CALLBACKS` 0U /\* DFSDM register callback disabled \*/
- #define `USE_HAL_DMA2D_REGISTER_CALLBACKS` 0U /\* DMA2D register callback disabled \*/
- #define `USE_HAL_DSI_REGISTER_CALLBACKS` 0U /\* DSI register callback disabled \*/
- #define `USE_HAL_ETH_REGISTER_CALLBACKS` 0U /\* ETH register callback disabled \*/
- #define `USE_HAL_HASH_REGISTER_CALLBACKS` 0U /\* HASH register callback disabled \*/
- #define `USE_HAL_HCD_REGISTER_CALLBACKS` 0U /\* HCD register callback disabled \*/
- #define `USE_HAL_I2C_REGISTER_CALLBACKS` 0U /\* I2C register callback disabled \*/
- #define `USE_HAL_FMPI2C_REGISTER_CALLBACKS` 0U /\* FMPI2C register callback disabled \*/
- #define `USE_HAL_FMPMBUS_REGISTER_CALLBACKS` 0U /\* FMPSMBUS register callback disabled \*/
- #define `USE_HAL_I2S_REGISTER_CALLBACKS` 0U /\* I2S register callback disabled \*/
- #define `USE_HAL_IRDA_REGISTER_CALLBACKS` 0U /\* IRDA register callback disabled \*/
- #define `USE_HAL_LPTIM_REGISTER_CALLBACKS` 0U /\* LPTIM register callback disabled \*/
- #define `USE_HAL_LTDC_REGISTER_CALLBACKS` 0U /\* LTDC register callback disabled \*/

- #define `USE_HAL_MMC_REGISTER_CALLBACKS` 0U /\* MMC register callback disabled \*/
- #define `USE_HAL_NAND_REGISTER_CALLBACKS` 0U /\* NAND register callback disabled \*/
- #define `USE_HAL_NOR_REGISTER_CALLBACKS` 0U /\* NOR register callback disabled \*/
- #define `USE_HAL_PCCARD_REGISTER_CALLBACKS` 0U /\* PCCARD register callback disabled \*/
- #define `USE_HAL_PCD_REGISTER_CALLBACKS` 0U /\* PCD register callback disabled \*/
- #define `USE_HAL_QSPI_REGISTER_CALLBACKS` 0U /\* QSPI register callback disabled \*/
- #define `USE_HAL_RNG_REGISTER_CALLBACKS` 0U /\* RNG register callback disabled \*/
- #define `USE_HAL_RTC_REGISTER_CALLBACKS` 0U /\* RTC register callback disabled \*/
- #define `USE_HAL_SAI_REGISTER_CALLBACKS` 0U /\* SAI register callback disabled \*/
- #define `USE_HAL_SD_REGISTER_CALLBACKS` 0U /\* SD register callback disabled \*/
- #define `USE_HAL_SMARTCARD_REGISTER_CALLBACKS` 0U /\* SMARTCARD register callback disabled \*/
- #define `USE_HAL_SDRAM_REGISTER_CALLBACKS` 0U /\* SDRAM register callback disabled \*/
- #define `USE_HAL_SRAM_REGISTER_CALLBACKS` 0U /\* SRAM register callback disabled \*/
- #define `USE_HAL_SPDIFRX_REGISTER_CALLBACKS` 0U /\* SPDIFRX register callback disabled \*/
- #define `USE_HAL_SMBUS_REGISTER_CALLBACKS` 0U /\* SMBUS register callback disabled \*/
- #define `USE_HAL_SPI_REGISTER_CALLBACKS` 0U /\* SPI register callback disabled \*/
- #define `USE_HAL_TIM_REGISTER_CALLBACKS` 0U /\* TIM register callback disabled \*/
- #define `USE_HAL_UART_REGISTER_CALLBACKS` 0U /\* UART register callback disabled \*/
- #define `USE_HAL_USART_REGISTER_CALLBACKS` 0U /\* USART register callback disabled \*/
- #define `USE_HAL_WWDG_REGISTER_CALLBACKS` 0U /\* WWDG register callback disabled \*/
- #define `MAC_ADDR0` 2U

*Uncomment the line below to expanse the "assert\_param" macro in the HAL drivers code.*

- #define `MAC_ADDR1` 0U
- #define `MAC_ADDR2` 0U
- #define `MAC_ADDR3` 0U
- #define `MAC_ADDR4` 0U
- #define `MAC_ADDR5` 0U
- #define `ETH_RX_BUF_SIZE` ETH\_MAX\_PACKET\_SIZE /\* buffer size for receive \*/
- #define `ETH_TX_BUF_SIZE` ETH\_MAX\_PACKET\_SIZE /\* buffer size for transmit \*/
- #define `ETH_RXBUFNB` 4U /\* 4 Rx buffers of size ETH\_RX\_BUF\_SIZE \*/
- #define `ETH_TXBUFNB` 4U /\* 4 Tx buffers of size ETH\_TX\_BUF\_SIZE \*/
- #define `DP83848_PHY_ADDRESS`
- #define `PHY_RESET_DELAY` 0x000000FFU
- #define `PHY_CONFIG_DELAY` 0x00000FFFU
- #define `PHY_READ_TO` 0x0000FFFFU
- #define `PHY_WRITE_TO` 0x0000FFFFU
- #define `PHY_BCR` ((uint16\_t)0x0000U)
- #define `PHY_BSR` ((uint16\_t)0x0001U)
- #define `PHY_RESET` ((uint16\_t)0x8000U)
- #define `PHY_LOOPBACK` ((uint16\_t)0x4000U)
- #define `PHY_FULLDUPLEX_100M` ((uint16\_t)0x2100U)
- #define `PHY_HALFDUPLEX_100M` ((uint16\_t)0x2000U)
- #define `PHY_FULLDUPLEX_10M` ((uint16\_t)0x0100U)
- #define `PHY_HALFDUPLEX_10M` ((uint16\_t)0x0000U)
- #define `PHY_AUTONEGOTIATION` ((uint16\_t)0x1000U)
- #define `PHY_RESTART_AUTONEGOTIATION` ((uint16\_t)0x0200U)
- #define `PHY_POWERDOWN` ((uint16\_t)0x0800U)
- #define `PHY_ISOLATE` ((uint16\_t)0x0400U)
- #define `PHY_AUTONEGO_COMPLETE` ((uint16\_t)0x0020U)
- #define `PHY_LINKED_STATUS` ((uint16\_t)0x0004U)
- #define `PHY_JABBER_DETECTION` ((uint16\_t)0x0002U)
- #define `PHY_SR` ((uint16\_t))
- #define `PHY_SPEED_STATUS` ((uint16\_t))
- #define `PHY_DUPLEX_STATUS` ((uint16\_t))
- #define `USE_SPI_CRC` 0U
- #define `assert_param`(expr) ((void)0U)

*Include module's header file.*

## 9.22.1 Macro Definition Documentation

### 9.22.1.1 assert\_param

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

Include module's header file.

### 9.22.1.2 DATA\_CACHE\_ENABLE

```
#define DATA_CACHE_ENABLE 1U
```

### 9.22.1.3 DP83848\_PHY\_ADDRESS

```
#define DP83848_PHY_ADDRESS
```

### 9.22.1.4 ETH\_RX\_BUF\_SIZE

```
#define ETH_RX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for receive */
```

### 9.22.1.5 ETH\_RXBUFNB

```
#define ETH_RXBUFNB 4U /* 4 Rx buffers of size ETH_RX_BUF_SIZE */
```

### 9.22.1.6 ETH\_TX\_BUF\_SIZE

```
#define ETH_TX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for transmit */
```

### 9.22.1.7 ETH\_TXBUFNB

```
#define ETH_TXBUFNB 4U /* 4 Tx buffers of size ETH_TX_BUF_SIZE */
```

### 9.22.1.8 EXTERNAL\_CLOCK\_VALUE

```
#define EXTERNAL_CLOCK_VALUE 12288000U
```

External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S\_CKIN pad.

Value of the External audio frequency in Hz

#### 9.22.1.9 HAL\_CORTEX\_MODULE\_ENABLED

```
#define HAL_CORTEX_MODULE_ENABLED
```

#### 9.22.1.10 HAL\_DMA\_MODULE\_ENABLED

```
#define HAL_DMA_MODULE_ENABLED
```

#### 9.22.1.11 HAL\_EXTI\_MODULE\_ENABLED

```
#define HAL_EXTI_MODULE_ENABLED
```

#### 9.22.1.12 HAL\_FLASH\_MODULE\_ENABLED

```
#define HAL_FLASH_MODULE_ENABLED
```

#### 9.22.1.13 HAL\_GPIO\_MODULE\_ENABLED

```
#define HAL_GPIO_MODULE_ENABLED
```

#### 9.22.1.14 HAL\_I2C\_MODULE\_ENABLED

```
#define HAL_I2C_MODULE_ENABLED
```

#### 9.22.1.15 HAL\_MODULE\_ENABLED

```
#define HAL_MODULE_ENABLED
```

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

#### 9.22.1.16 HAL\_PWR\_MODULE\_ENABLED

```
#define HAL_PWR_MODULE_ENABLED
```

#### 9.22.1.17 HAL\_RCC\_MODULE\_ENABLED

```
#define HAL_RCC_MODULE_ENABLED
```

#### 9.22.1.18 HAL\_UART\_MODULE\_ENABLED

```
#define HAL_UART_MODULE_ENABLED
```



#### 9.22.1.19 HSE\_STARTUP\_TIMEOUT

```
#define HSE_STARTUP_TIMEOUT 100U
```

Time out for HSE start up, in ms

#### 9.22.1.20 HSE\_VALUE

```
#define HSE_VALUE 25000000U
```

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

#### 9.22.1.21 HSI\_VALUE

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

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

#### 9.22.1.22 INSTRUCTION\_CACHE\_ENABLE

```
#define INSTRUCTION_CACHE_ENABLE 1U
```

#### 9.22.1.23 LSE\_STARTUP\_TIMEOUT

```
#define LSE_STARTUP_TIMEOUT 5000U
```

Time out for LSE start up, in ms

#### 9.22.1.24 LSE\_VALUE

```
#define LSE_VALUE 32768U
```

External Low Speed oscillator (LSE) value.

< 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 Low Speed oscillator in Hz

#### 9.22.1.25 LSI\_VALUE

```
#define LSI_VALUE 32000U
```

Internal Low Speed oscillator (LSI) value.

LSI Typical Value in Hz

#### 9.22.1.26 MAC\_ADDR0

```
#define MAC_ADDR0 2U
```

Uncomment the line below to expanse the "assert\_param" macro in the HAL drivers code.

#### 9.22.1.27 MAC\_ADDR1

```
#define MAC_ADDR1 0U
```

#### 9.22.1.28 MAC\_ADDR2

```
#define MAC_ADDR2 0U
```

#### 9.22.1.29 MAC\_ADDR3

```
#define MAC_ADDR3 0U
```

#### 9.22.1.30 MAC\_ADDR4

```
#define MAC_ADDR4 0U
```

#### 9.22.1.31 MAC\_ADDR5

```
#define MAC_ADDR5 0U
```

#### 9.22.1.32 PHY\_AUTONEGO\_COMPLETE

```
#define PHY_AUTONEGO_COMPLETE ((uint16_t)0x0020U)
```

Auto-Negotiation process completed

### 9.22.1.33 PHY\_AUTONEGOTIATION

```
#define PHY_AUTONEGOTIATION ((uint16_t)0x1000U)
```

Enable auto-negotiation function

### 9.22.1.34 PHY\_BCR

```
#define PHY_BCR ((uint16_t)0x0000U)
```

Transceiver Basic Control Register

### 9.22.1.35 PHY\_BSR

```
#define PHY_BSR ((uint16_t)0x0001U)
```

Transceiver Basic Status Register

### 9.22.1.36 PHY\_CONFIG\_DELAY

```
#define PHY_CONFIG_DELAY 0x00000FFFU
```

### 9.22.1.37 PHY\_DUPLEX\_STATUS

```
#define PHY_DUPLEX_STATUS ((uint16_t))
```

PHY Duplex mask

### 9.22.1.38 PHY\_FULLDUPLEX\_100M

```
#define PHY_FULLDUPLEX_100M ((uint16_t)0x2100U)
```

Set the full-duplex mode at 100 Mb/s

### 9.22.1.39 PHY\_FULLDUPLEX\_10M

```
#define PHY_FULLDUPLEX_10M ((uint16_t)0x0100U)
```

Set the full-duplex mode at 10 Mb/s

#### 9.22.1.40 PHY\_HALFDUPLEX\_100M

```
#define PHY_HALFDUPLEX_100M ((uint16_t)0x2000U)
```

Set the half-duplex mode at 100 Mb/s

#### 9.22.1.41 PHY\_HALFDUPLEX\_10M

```
#define PHY_HALFDUPLEX_10M ((uint16_t)0x0000U)
```

Set the half-duplex mode at 10 Mb/s

#### 9.22.1.42 PHY\_ISOLATE

```
#define PHY_ISOLATE ((uint16_t)0x0400U)
```

Isolate PHY from MII

#### 9.22.1.43 PHY\_JABBER\_DETECTION

```
#define PHY_JABBER_DETECTION ((uint16_t)0x0002U)
```

Jabber condition detected

#### 9.22.1.44 PHY\_LINKED\_STATUS

```
#define PHY_LINKED_STATUS ((uint16_t)0x0004U)
```

Valid link established

#### 9.22.1.45 PHY\_LOOPBACK

```
#define PHY_LOOPBACK ((uint16_t)0x4000U)
```

Select loop-back mode

#### 9.22.1.46 PHY\_POWERDOWN

```
#define PHY_POWERDOWN ((uint16_t)0x0800U)
```

Select the power down mode

#### 9.22.1.47 PHY\_READ\_TO

```
#define PHY_READ_TO 0x0000FFFFU
```

#### 9.22.1.48 PHY\_RESET

```
#define PHY_RESET ((uint16_t)0x8000U)
```

PHY Reset

#### 9.22.1.49 PHY\_RESET\_DELAY

```
#define PHY_RESET_DELAY 0x000000FFU
```

#### 9.22.1.50 PHY\_RESTART\_AUTONEGOTIATION

```
#define PHY_RESTART_AUTONEGOTIATION ((uint16_t)0x0200U)
```

Restart auto-negotiation function

#### 9.22.1.51 PHY\_SPEED\_STATUS

```
#define PHY_SPEED_STATUS ((uint16_t))
```

PHY Speed mask

#### 9.22.1.52 PHY\_SR

```
#define PHY_SR ((uint16_t))
```

PHY status register Offset

#### 9.22.1.53 PHY\_WRITE\_TO

```
#define PHY_WRITE_TO 0x0000FFFFU
```

#### 9.22.1.54 PREFETCH\_ENABLE

```
#define PREFETCH_ENABLE 1U
```

#### 9.22.1.55 TICK\_INT\_PRIORITY

```
#define TICK_INT_PRIORITY 15U
```

tick interrupt priority

#### 9.22.1.56 USE\_HAL\_ADC\_REGISTER\_CALLBACKS

```
#define USE_HAL_ADC_REGISTER_CALLBACKS 0U /* ADC register callback disabled */
```

#### 9.22.1.57 USE\_HAL\_CAN\_REGISTER\_CALLBACKS

```
#define USE_HAL_CAN_REGISTER_CALLBACKS 0U /* CAN register callback disabled */
```

#### 9.22.1.58 USE\_HAL\_CEC\_REGISTER\_CALLBACKS

```
#define USE_HAL_CEC_REGISTER_CALLBACKS 0U /* CEC register callback disabled */
```

#### 9.22.1.59 USE\_HAL\_Cryp\_REGISTER\_CALLBACKS

```
#define USE_HAL_Cryp_REGISTER_CALLBACKS 0U /* CRYP register callback disabled */
```

#### 9.22.1.60 USE\_HAL\_DAC\_REGISTER\_CALLBACKS

```
#define USE_HAL_DAC_REGISTER_CALLBACKS 0U /* DAC register callback disabled */
```

#### 9.22.1.61 USE\_HAL\_DCMI\_REGISTER\_CALLBACKS

```
#define USE_HAL_DCMI_REGISTER_CALLBACKS 0U /* DCMI register callback disabled */
```

#### 9.22.1.62 USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS

```
#define USE_HAL_DFSDM_REGISTER_CALLBACKS 0U /* DFSDM register callback disabled */
```

#### 9.22.1.63 USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS

```
#define USE_HAL_DMA2D_REGISTER_CALLBACKS 0U /* DMA2D register callback disabled */
```

#### 9.22.1.64 USE\_HAL\_DSI\_REGISTER\_CALLBACKS

```
#define USE_HAL_DSI_REGISTER_CALLBACKS 0U /* DSI register callback disabled */
```

#### 9.22.1.65 USE\_HAL\_ETH\_REGISTER\_CALLBACKS

```
#define USE_HAL_ETH_REGISTER_CALLBACKS 0U /* ETH register callback disabled */
```

#### 9.22.1.66 USE\_HAL\_FMPI2C\_REGISTER\_CALLBACKS

```
#define USE_HAL_FMPI2C_REGISTER_CALLBACKS 0U /* FMPI2C register callback disabled */
```

#### 9.22.1.67 USE\_HAL\_FMPMBUS\_REGISTER\_CALLBACKS

```
#define USE_HAL_FMPMBUS_REGISTER_CALLBACKS 0U /* FMPMBUS register callback disabled */
```

#### 9.22.1.68 USE\_HAL\_HASH\_REGISTER\_CALLBACKS

```
#define USE_HAL_HASH_REGISTER_CALLBACKS 0U /* HASH register callback disabled */
```

#### 9.22.1.69 USE\_HAL\_HCD\_REGISTER\_CALLBACKS

```
#define USE_HAL_HCD_REGISTER_CALLBACKS 0U /* HCD register callback disabled */
```

#### 9.22.1.70 USE\_HAL\_I2C\_REGISTER\_CALLBACKS

```
#define USE_HAL_I2C_REGISTER_CALLBACKS 0U /* I2C register callback disabled */
```

#### 9.22.1.71 USE\_HAL\_I2S\_REGISTER\_CALLBACKS

```
#define USE_HAL_I2S_REGISTER_CALLBACKS 0U /* I2S register callback disabled */
```

#### 9.22.1.72 USE\_HAL\_IRDA\_REGISTER\_CALLBACKS

```
#define USE_HAL_IRDA_REGISTER_CALLBACKS 0U /* IRDA register callback disabled */
```

#### 9.22.1.73 USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS

```
#define USE_HAL_LPTIM_REGISTER_CALLBACKS 0U /* LPTIM register callback disabled */
```

#### 9.22.1.74 USE\_HAL\_LTDC\_REGISTER\_CALLBACKS

```
#define USE_HAL_LTDC_REGISTER_CALLBACKS 0U /* LTDC register callback disabled */
```

#### 9.22.1.75 USE\_HAL\_MMC\_REGISTER\_CALLBACKS

```
#define USE_HAL_MMC_REGISTER_CALLBACKS 0U /* MMC register callback disabled */
```

#### 9.22.1.76 USE\_HAL\_NAND\_REGISTER\_CALLBACKS

```
#define USE_HAL_NAND_REGISTER_CALLBACKS 0U /* NAND register callback disabled */
```

#### 9.22.1.77 USE\_HAL\_NOR\_REGISTER\_CALLBACKS

```
#define USE_HAL_NOR_REGISTER_CALLBACKS 0U /* NOR register callback disabled */
```

#### 9.22.1.78 USE\_HAL\_PCCARD\_REGISTER\_CALLBACKS

```
#define USE_HAL_PCCARD_REGISTER_CALLBACKS 0U /* PCCARD register callback disabled */
```

#### 9.22.1.79 USE\_HAL\_PCD\_REGISTER\_CALLBACKS

```
#define USE_HAL_PCD_REGISTER_CALLBACKS 0U /* PCD register callback disabled */
```

#### 9.22.1.80 USE\_HAL\_QSPI\_REGISTER\_CALLBACKS

```
#define USE_HAL_QSPI_REGISTER_CALLBACKS 0U /* QSPI register callback disabled */
```

#### 9.22.1.81 USE\_HAL\_RNG\_REGISTER\_CALLBACKS

```
#define USE_HAL_RNG_REGISTER_CALLBACKS 0U /* RNG register callback disabled */
```

#### 9.22.1.82 USE\_HAL\_RTC\_REGISTER\_CALLBACKS

```
#define USE_HAL_RTC_REGISTER_CALLBACKS 0U /* RTC register callback disabled */
```

#### 9.22.1.83 USE\_HAL\_SAI\_REGISTER\_CALLBACKS

```
#define USE_HAL_SAI_REGISTER_CALLBACKS 0U /* SAI register callback disabled */
```

#### 9.22.1.84 USE\_HAL\_SD\_REGISTER\_CALLBACKS

```
#define USE_HAL_SD_REGISTER_CALLBACKS 0U /* SD register callback disabled */
```



**9.22.1.85 USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SDRAM_REGISTER_CALLBACKS 0U /* SDRAM register callback disabled */
```

**9.22.1.86 USE\_HAL\_SMARTCARD\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SMARTCARD_REGISTER_CALLBACKS 0U /* SMARTCARD register callback disabled */
```

**9.22.1.87 USE\_HAL\_SMBUS\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SMBUS_REGISTER_CALLBACKS 0U /* SMBUS register callback disabled */
```

**9.22.1.88 USE\_HAL\_SPDIFRX\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SPDIFRX_REGISTER_CALLBACKS 0U /* SPDIFRX register callback disabled */
```

**9.22.1.89 USE\_HAL\_SPI\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SPI_REGISTER_CALLBACKS 0U /* SPI register callback disabled */
```

**9.22.1.90 USE\_HAL\_SRAM\_REGISTER\_CALLBACKS**

```
#define USE_HAL_SRAM_REGISTER_CALLBACKS 0U /* SRAM register callback disabled */
```

**9.22.1.91 USE\_HAL\_TIM\_REGISTER\_CALLBACKS**

```
#define USE_HAL_TIM_REGISTER_CALLBACKS 0U /* TIM register callback disabled */
```

**9.22.1.92 USE\_HAL\_UART\_REGISTER\_CALLBACKS**

```
#define USE_HAL_UART_REGISTER_CALLBACKS 0U /* UART register callback disabled */
```

**9.22.1.93 USE\_HAL\_USART\_REGISTER\_CALLBACKS**

```
#define USE_HAL_USART_REGISTER_CALLBACKS 0U /* USART register callback disabled */
```

**9.22.1.94 USE\_HAL\_WWDG\_REGISTER\_CALLBACKS**

```
#define USE_HAL_WWDG_REGISTER_CALLBACKS 0U /* WWDG register callback disabled */
```

### 9.22.1.95 USE\_RTOS

```
#define USE_RTOS 0U
```

### 9.22.1.96 USE\_SPI\_CRC

```
#define USE_SPI_CRC 0U
```

### 9.22.1.97 VDD\_VALUE

```
#define VDD_VALUE 3300U
```

This is the HAL system configuration section.

Value of VDD in mv

## 9.23 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/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 [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.*

### 9.23.1 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.

### 9.23.2 Function Documentation

#### 9.23.2.1 BusFault\_Handler()

```
void BusFault_Handler (
    void )
```

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

#### 9.23.2.2 DebugMon\_Handler()

```
void DebugMon_Handler (
    void )
```

This function handles Debug monitor.

#### 9.23.2.3 HardFault\_Handler()

```
void HardFault_Handler (
    void )
```

This function handles Hard fault interrupt.

#### 9.23.2.4 MemManage\_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

#### 9.23.2.5 NMI\_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

### 9.23.2.6 PendSV\_Handler()

```
void PendSV_Handler (
    void )
```

This function handles Pendable request for system service.

### 9.23.2.7 SVC\_Handler()

```
void SVC_Handler (
    void )
```

This function handles System service call via SWI instruction.

### 9.23.2.8 SysTick\_Handler()

```
void SysTick_Handler (
    void )
```

This function handles System tick timer.

### 9.23.2.9 UsageFault\_Handler()

```
void UsageFault_Handler (
    void )
```

This function handles Undefined instruction or illegal state.

## 9.24 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/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 [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 [TIM4\\_IRQHandler](#) (void)
- void [USART1\\_IRQHandler](#) (void)

### 9.24.1 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.

### 9.24.2 Function Documentation

#### 9.24.2.1 BusFault\_Handler()

```
void BusFault_Handler (
    void )
```

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

#### 9.24.2.2 DebugMon\_Handler()

```
void DebugMon_Handler (
    void )
```

This function handles Debug monitor.

#### 9.24.2.3 HardFault\_Handler()

```
void HardFault_Handler (
    void )
```

This function handles Hard fault interrupt.

#### 9.24.2.4 MemManage\_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

#### 9.24.2.5 NMI\_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

#### 9.24.2.6 PendSV\_Handler()

```
void PendSV_Handler (
    void )
```

This function handles Pendable request for system service.

#### 9.24.2.7 SVC\_Handler()

```
void SVC_Handler (
    void )
```

This function handles System service call via SWI instruction.

#### 9.24.2.8 SysTick\_Handler()

```
void SysTick_Handler (
    void )
```

This function handles System tick timer.

#### 9.24.2.9 TIM4\_IRQHandler()

```
void TIM4_IRQHandler (
    void )
```

#### 9.24.2.10 UsageFault\_Handler()

```
void UsageFault_Handler (
    void )
```

This function handles Undefined instruction or illegal state.

#### 9.24.2.11 USART1\_IRQHandler()

```
void USART1_IRQHandler (
    void )
```

## 9.25 Ball\_Launcher\_Controller/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 [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.*

### 9.25.1 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.

### 9.25.2 Function Documentation

#### 9.25.2.1 BusFault\_Handler()

```
void BusFault_Handler (  
    void )
```

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

#### 9.25.2.2 DebugMon\_Handler()

```
void DebugMon_Handler (  
    void )
```

This function handles Debug monitor.

#### 9.25.2.3 HardFault\_Handler()

```
void HardFault_Handler (
    void )
```

This function handles Hard fault interrupt.

#### 9.25.2.4 MemManage\_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

#### 9.25.2.5 NMI\_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

#### 9.25.2.6 PendSV\_Handler()

```
void PendSV_Handler (
    void )
```

This function handles Pendable request for system service.

#### 9.25.2.7 SVC\_Handler()

```
void SVC_Handler (
    void )
```

This function handles System service call via SWI instruction.

#### 9.25.2.8 SysTick\_Handler()

```
void SysTick_Handler (
    void )
```

This function handles System tick timer.

#### 9.25.2.9 UsageFault\_Handler()

```
void UsageFault_Handler (
    void )
```

This function handles Undefined instruction or illegal state.



## 9.26 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/↵ Src/main.c File Reference

Main program body.

```
#include "main.h"  
#include "mpu6050.h"  
#include <stdio.h>  
#include <string.h>
```

### Functions

- void [SystemClock\\_Config](#) (void)  
*System Clock Configuration.*
- int [main](#) (void)
- void [Error\\_Handler](#) (void)  
*This function is executed in case of error occurrence.*

### Variables

- I2C\_HandleTypeDef [hi2c1](#)
- UART\_HandleTypeDef [huart1](#)
- UART\_HandleTypeDef [huart2](#)
- int32\_t [gX](#) = 0
- int32\_t [gY](#) = 0
- int32\_t [gZ](#) = 0
- uint16\_t [stat](#) = 0
- uint16\_t [shot](#) = 0
- uint16\_t [move](#) = 0
- char [buffer](#) [100]

### 9.26.1 Detailed Description

Main program body.

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.

### 9.26.2 Function Documentation

#### 9.26.2.1 Error\_Handler()

```
void Error_Handler (  
    void )
```

This function is executed in case of error occurrence.

**Return values**

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

**9.26.2.2 main()**

```
int main (
        void )
```

**9.26.2.3 SystemClock\_Config()**

```
void SystemClock_Config (
        void )
```

System Clock Configuration.

**Return values**

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

Configure the main internal regulator output voltage

Initializes the RCC Oscillators according to the specified parameters in the RCC\_OscInitTypeDef structure.

Initializes the CPU, AHB and APB buses clocks

**9.26.3 Variable Documentation****9.26.3.1 buffer**

```
char buffer[100]
```

**9.26.3.2 gX**

```
int32_t gX = 0
```

**9.26.3.3 gY**

```
int32_t gY = 0
```

**9.26.3.4 gZ**

```
int32_t gZ = 0
```

### 9.26.3.5 hi2c1

```
I2C_HandleTypeDef hi2c1
```

### 9.26.3.6 huart1

```
UART_HandleTypeDef huart1
```

### 9.26.3.7 huart2

```
UART_HandleTypeDef huart2
```

### 9.26.3.8 move

```
uint16_t move = 0
```

### 9.26.3.9 shot

```
uint16_t shot = 0
```

### 9.26.3.10 stat

```
uint16_t stat = 0
```

## 9.27 Ball\_Launcher\_Controller/Core/Src/main.c File Reference

Main program body.

```
#include "main.h"  
#include "mpu6050.h"  
#include <stdio.h>  
#include <string.h>
```

### Functions

- void [SystemClock\\_Config](#) (void)  
*System Clock Configuration.*
- int [main](#) (void)
- void [Error\\_Handler](#) (void)  
*This function is executed in case of error occurrence.*

## Variables

- I2C\_HandleTypeDef [hi2c1](#)
- UART\_HandleTypeDef [huart1](#)
- UART\_HandleTypeDef [huart2](#)
- int32\_t [gX](#) = 0
- int32\_t [gY](#) = 0
- int32\_t [gZ](#) = 0
- uint16\_t [stat](#) = 0
- uint16\_t [shot](#) = 0
- uint16\_t [move](#) = 0
- char [buffer](#) [100]

### 9.27.1 Detailed Description

Main program body.

#### 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.

### 9.27.2 Function Documentation

#### 9.27.2.1 Error\_Handler()

```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

#### Return values

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

#### 9.27.2.2 main()

```
int main (
    void )
```

#### 9.27.2.3 SystemClock\_Config()

```
void SystemClock_Config (
    void )
```

System Clock Configuration.

**Return values**

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

Configure the main internal regulator output voltage

Initializes the RCC Oscillators according to the specified parameters in the RCC\_OscInitTypeDef structure.

Initializes the CPU, AHB and APB buses clocks

### 9.27.3 Variable Documentation

#### 9.27.3.1 buffer

```
char buffer[100]
```

#### 9.27.3.2 gX

```
int32_t gX = 0
```

#### 9.27.3.3 gY

```
int32_t gY = 0
```

#### 9.27.3.4 gZ

```
int32_t gZ = 0
```

#### 9.27.3.5 hi2c1

```
I2C_HandleTypeDef hi2c1
```

#### 9.27.3.6 huart1

```
UART_HandleTypeDef huart1
```

#### 9.27.3.7 huart2

```
UART_HandleTypeDef huart2
```

#### 9.27.3.8 move

```
uint16_t move = 0
```

### 9.27.3.9 shot

```
uint16_t shot = 0
```

### 9.27.3.10 stat

```
uint16_t stat = 0
```

## 9.28 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/↵ Src/mpu6050.c File Reference

Implementation of mpu6050 sensor functions.

```
#include "mpu6050.h"
```

### Functions

- `uint16_t mpu6050_init (MPU6050 *imux, I2C_HandleTypeDef *hi2c)`  
*Initializes the MPU6050 sensor.*
- `void mpu6050_calibrate (MPU6050 *imux)`  
*Calibrates the MPU6050 sensor.*
- `void mpu6050_update (MPU6050 *imux)`  
*Updates the MPU6050 sensor data.*
- `int16_t mpu6050_get_gX (MPU6050 *imux)`  
*Gets the calibrated X-axis gyroscope data.*
- `int16_t mpu6050_get_gY (MPU6050 *imux)`  
*Gets the calibrated Y-axis gyroscope data.*
- `int16_t mpu6050_get_gZ (MPU6050 *imux)`  
*Gets the calibrated Z-axis gyroscope data.*

### 9.28.1 Detailed Description

Implementation of mpu6050 sensor functions.

Created on: May 3, 2024 Author: vvinh

### 9.28.2 Function Documentation

#### 9.28.2.1 mpu6050\_calibrate()

```
void mpu6050_calibrate (  
    MPU6050 * imux )
```

Calibrates the MPU6050 sensor.

**Parameters**

<i>imax</i>	Pointer to the <a href="#">MPU6050</a> structure.
-------------	---

**9.28.2.2 mpu6050\_get\_gX()**

```
int16_t mpu6050_get_gX (
    MPU6050 * imax )
```

Gets the calibrated X-axis gyroscope data.

**Parameters**

<i>imax</i>	Pointer to the <a href="#">MPU6050</a> structure.
-------------	---

**Returns**

int16\_t Calibrated X-axis gyroscope data.

**9.28.2.3 mpu6050\_get\_gY()**

```
int16_t mpu6050_get_gY (
    MPU6050 * imax )
```

Gets the calibrated Y-axis gyroscope data.

**Parameters**

<i>imax</i>	Pointer to the <a href="#">MPU6050</a> structure.
-------------	---

**Returns**

int16\_t Calibrated Y-axis gyroscope data.

**9.28.2.4 mpu6050\_get\_gZ()**

```
int16_t mpu6050_get_gZ (
    MPU6050 * imax )
```

Gets the calibrated Z-axis gyroscope data.

**Parameters**

<i>imax</i>	Pointer to the <a href="#">MPU6050</a> structure.
-------------	---



**Returns**

int16\_t Calibrated Z-axis gyroscope data.

**9.28.2.5 mpu6050\_init()**

```
uint16_t mpu6050_init (
    MPU6050 * imux,
    I2C_HandleTypeDef * hi2c )
```

Initializes the MPU6050 sensor.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
<i>hi2c</i>	Pointer to the I2C handle structure.

**Returns**

uint16\_t Returns 1 if initialization is successful, otherwise 0.

**9.28.2.6 mpu6050\_update()**

```
void mpu6050_update (
    MPU6050 * imux )
```

Updates the MPU6050 sensor data.

**Parameters**

<i>imux</i>	Pointer to the MPU6050 structure.
<i>dt</i>	Time interval in milliseconds.

## 9.29 Ball\_Launcher\_Controller/Core/Src/mpu6050.c File Reference

Implementation of mpu6050 sensor functions.

```
#include "mpu6050.h"
```

**Functions**

- uint16\_t **mpu6050\_init** (MPU6050 \*imux, I2C\_HandleTypeDef \*hi2c)  
*Initializes the MPU6050 sensor.*
- void **mpu6050\_calibrate** (MPU6050 \*imux)  
*Calibrates the MPU6050 sensor.*

- void `mpu6050_update` (`MPU6050 *imux`)  
*Updates the MPU6050 sensor data.*
- int16\_t `mpu6050_get_gX` (`MPU6050 *imux`)  
*Gets the calibrated X-axis gyroscope data.*
- int16\_t `mpu6050_get_gY` (`MPU6050 *imux`)  
*Gets the calibrated Y-axis gyroscope data.*
- int16\_t `mpu6050_get_gZ` (`MPU6050 *imux`)  
*Gets the calibrated Z-axis gyroscope data.*

### 9.29.1 Detailed Description

Implementation of mpu6050 sensor functions.

Created on: May 3, 2024 Author: vvinh

### 9.29.2 Function Documentation

#### 9.29.2.1 mpu6050\_calibrate()

```
void mpu6050_calibrate (
    MPU6050 * imux )
```

Calibrates the MPU6050 sensor.

##### Parameters

<code>imux</code>	Pointer to the MPU6050 structure.
-------------------	-----------------------------------

#### 9.29.2.2 mpu6050\_get\_gX()

```
int16_t mpu6050_get_gX (
    MPU6050 * imux )
```

Gets the calibrated X-axis gyroscope data.

##### Parameters

<code>imux</code>	Pointer to the MPU6050 structure.
-------------------	-----------------------------------

##### Returns

int16\_t Calibrated X-axis gyroscope data.

#### 9.29.2.3 mpu6050\_get\_gY()

```
int16_t mpu6050_get_gY (
    MPU6050 * imux )
```

Gets the calibrated Y-axis gyroscope data.

#### Parameters

<i>imux</i>	Pointer to the <a href="#">MPU6050</a> structure.
-------------	---

#### Returns

int16\_t Calibrated Y-axis gyroscope data.

### 9.29.2.4 mpu6050\_get\_gZ()

```
int16_t mpu6050_get_gZ (  
    MPU6050 * imux )
```

Gets the calibrated Z-axis gyroscope data.

#### Parameters

<i>imux</i>	Pointer to the <a href="#">MPU6050</a> structure.
-------------	---

#### Returns

int16\_t Calibrated Z-axis gyroscope data.

### 9.29.2.5 mpu6050\_init()

```
uint16_t mpu6050_init (  
    MPU6050 * imux,  
    I2C_HandleTypeDef * hi2c )
```

Initializes the [MPU6050](#) sensor.

#### Parameters

<i>imux</i>	Pointer to the <a href="#">MPU6050</a> structure.
<i>hi2c</i>	Pointer to the I2C handle structure.

#### Returns

uint16\_t Returns 1 if initialization is successful, otherwise 0.

### 9.29.2.6 mpu6050\_update()

```
void mpu6050_update (  
    MPU6050 * imux )
```

Updates the [MPU6050](#) sensor data.

## Parameters

<i>imux</i>	Pointer to the <a href="#">MPU6050</a> structure.
<i>dt</i>	Time interval in milliseconds.

### 9.30 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/↵ Src/stm32f4xx\_hal\_msp.c File Reference

This file provides code for the MSP Initialization and de-Initialization codes.

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

## Functions

- void [HAL\\_MspInit](#) (void)
- void [HAL\\_I2C\\_MspInit](#) (I2C\_HandleTypeDef \*hi2c)  
*I2C MSP Initialization This function configures the hardware resources used in this example.*
- void [HAL\\_I2C\\_MspDeInit](#) (I2C\_HandleTypeDef \*hi2c)  
*I2C MSP De-Initialization This function freeze the hardware resources used in this example.*
- void [HAL\\_UART\\_MspInit](#) (UART\_HandleTypeDef \*huart)  
*UART MSP Initialization This function configures the hardware resources used in this example.*
- void [HAL\\_UART\\_MspDeInit](#) (UART\_HandleTypeDef \*huart)  
*UART MSP De-Initialization This function freeze the hardware resources used in this example.*

#### 9.30.1 Detailed Description

This file provides code for the MSP Initialization and de-Initialization codes.

## 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.

#### 9.30.2 Function Documentation

##### 9.30.2.1 HAL\_I2C\_MspDeInit()

```
void HAL_I2C_MspDeInit (
    I2C_HandleTypeDef * hi2c )
```

I2C MSP De-Initialization This function freeze the hardware resources used in this example.

**Parameters**

<i>hi2c</i>	I2C handle pointer
-------------	--------------------

**Return values**

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

I2C1 GPIO Configuration PB6 -----> I2C1\_SCL PB7 -----> I2C1\_SDA

**9.30.2.2 HAL\_I2C\_MspInit()**

```
void HAL_I2C_MspInit (
    I2C_HandleTypeDef * hi2c )
```

I2C MSP Initialization This function configures the hardware resources used in this example.

**Parameters**

<i>hi2c</i>	I2C handle pointer
-------------	--------------------

**Return values**

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

I2C1 GPIO Configuration PB6 -----> I2C1\_SCL PB7 -----> I2C1\_SDA

**9.30.2.3 HAL\_MspInit()**

```
void HAL_MspInit (
    void )
```

Initializes the Global MSP.

**9.30.2.4 HAL\_UART\_MspDeInit()**

```
void HAL_UART_MspDeInit (
    UART_HandleTypeDef * huart )
```

UART MSP De-Initialization This function freeze the hardware resources used in this example.

**Parameters**

<i>huart</i>	UART handle pointer
--------------	---------------------

## Return values

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

USART1 GPIO Configuration PA9 -----> USART1\_TX PA10 -----> USART1\_RX

USART2 GPIO Configuration PA2 -----> USART2\_TX PA3 -----> USART2\_RX

### 9.30.2.5 HAL\_UART\_MspInit()

```
void HAL_UART_MspInit (
    UART_HandleTypeDef * huart )
```

UART MSP Initialization This function configures the hardware resources used in this example.

## Parameters

<i>huart</i>	UART handle pointer
--------------	---------------------

## Return values

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

USART1 GPIO Configuration PA9 -----> USART1\_TX PA10 -----> USART1\_RX

USART2 GPIO Configuration PA2 -----> USART2\_TX PA3 -----> USART2\_RX

## 9.31 Ball\_Launcher\_Controller/Core/Src/stm32f4xx\_hal\_msp.c File Reference

This file provides code for the MSP Initialization and de-Initialization codes.

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

## Functions

- void [HAL\\_MspInit](#) (void)
- void [HAL\\_I2C\\_MspInit](#) (I2C\_HandleTypeDef \*hi2c)  
*I2C MSP Initialization This function configures the hardware resources used in this example.*
- void [HAL\\_I2C\\_MspDeInit](#) (I2C\_HandleTypeDef \*hi2c)  
*I2C MSP De-Initialization This function freeze the hardware resources used in this example.*
- void [HAL\\_UART\\_MspInit](#) (UART\_HandleTypeDef \*huart)  
*UART MSP Initialization This function configures the hardware resources used in this example.*
- void [HAL\\_UART\\_MspDeInit](#) (UART\_HandleTypeDef \*huart)  
*UART MSP De-Initialization This function freeze the hardware resources used in this example.*

### 9.31.1 Detailed Description

This file provides code for the MSP Initialization and de-Initialization codes.

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.

### 9.31.2 Function Documentation

#### 9.31.2.1 HAL\_I2C\_MspDeInit()

```
void HAL_I2C_MspDeInit (
    I2C_HandleTypeDef * hi2c )
```

I2C MSP De-Initialization This function freeze the hardware resources used in this example.

Parameters

<i>hi2c</i>	I2C handle pointer
-------------	--------------------

Return values

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

I2C1 GPIO Configuration PB6 -----> I2C1\_SCL PB7 -----> I2C1\_SDA

#### 9.31.2.2 HAL\_I2C\_MspInit()

```
void HAL_I2C_MspInit (
    I2C_HandleTypeDef * hi2c )
```

I2C MSP Initialization This function configures the hardware resources used in this example.

Parameters

<i>hi2c</i>	I2C handle pointer
-------------	--------------------

Return values

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

I2C1 GPIO Configuration PB6 -----> I2C1\_SCL PB7 -----> I2C1\_SDA

### 9.31.2.3 HAL\_MspInit()

```
void HAL_MspInit (
    void )
```

Initializes the Global MSP.

### 9.31.2.4 HAL\_UART\_MspDeInit()

```
void HAL_UART_MspDeInit (
    UART_HandleTypeDef * huart )
```

UART MSP De-Initialization This function freeze the hardware resources used in this example.

#### Parameters

<i>huart</i>	UART handle pointer
--------------	---------------------

#### Return values

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

USART1 GPIO Configuration PA9 -----> USART1\_TX PA10 -----> USART1\_RX

USART2 GPIO Configuration PA2 -----> USART2\_TX PA3 -----> USART2\_RX

### 9.31.2.5 HAL\_UART\_MspInit()

```
void HAL_UART_MspInit (
    UART_HandleTypeDef * huart )
```

UART MSP Initialization This function configures the hardware resources used in this example.

#### Parameters

<i>huart</i>	UART handle pointer
--------------	---------------------

#### Return values

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

USART1 GPIO Configuration PA9 -----> USART1\_TX PA10 -----> USART1\_RX

USART2 GPIO Configuration PA2 -----> USART2\_TX PA3 -----> USART2\_RX



## 9.32 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/↵ Src/stm32f4xx\_it.c File Reference

Interrupt Service Routines.

```
#include "main.h"  
#include "stm32f4xx_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.*

### 9.32.1 Detailed Description

Interrupt Service Routines.

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.

### 9.32.2 Function Documentation

#### 9.32.2.1 BusFault\_Handler()

```
void BusFault_Handler (  
    void )
```

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

#### 9.32.2.2 DebugMon\_Handler()

```
void DebugMon_Handler (  
    void )
```

This function handles Debug monitor.

#### 9.32.2.3 HardFault\_Handler()

```
void HardFault_Handler (  
    void )
```

This function handles Hard fault interrupt.

#### 9.32.2.4 MemManage\_Handler()

```
void MemManage_Handler (  
    void )
```

This function handles Memory management fault.

#### 9.32.2.5 NMI\_Handler()

```
void NMI_Handler (  
    void )
```

This function handles Non maskable interrupt.

#### 9.32.2.6 PendSV\_Handler()

```
void PendSV_Handler (  
    void )
```

This function handles Pendable request for system service.

#### 9.32.2.7 SVC\_Handler()

```
void SVC_Handler (  
    void )
```

This function handles System service call via SWI instruction.

#### 9.32.2.8 SysTick\_Handler()

```
void SysTick_Handler (  
    void )
```

This function handles System tick timer.

### 9.32.2.9 UsageFault\_Handler()

```
void UsageFault_Handler (
    void )
```

This function handles Undefined instruction or illegal state.

## 9.33 Ball\_Launcher\_Controller/Core/Src/stm32f4xx\_it.c File Reference

Interrupt Service Routines.

```
#include "main.h"
#include "stm32f4xx_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.*

### 9.33.1 Detailed Description

Interrupt Service Routines.

#### 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.

## 9.33.2 Function Documentation

### 9.33.2.1 BusFault\_Handler()

```
void BusFault_Handler (
    void )
```

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

### 9.33.2.2 DebugMon\_Handler()

```
void DebugMon_Handler (
    void )
```

This function handles Debug monitor.

### 9.33.2.3 HardFault\_Handler()

```
void HardFault_Handler (
    void )
```

This function handles Hard fault interrupt.

### 9.33.2.4 MemManage\_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

### 9.33.2.5 NMI\_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

### 9.33.2.6 PendSV\_Handler()

```
void PendSV_Handler (
    void )
```

This function handles Pendable request for system service.

### 9.33.2.7 SVC\_Handler()

```
void SVC_Handler (
    void )
```

This function handles System service call via SWI instruction.

### 9.33.2.8 SysTick\_Handler()

```
void SysTick_Handler (
    void )
```

This function handles System tick timer.

### 9.33.2.9 UsageFault\_Handler()

```
void UsageFault_Handler (
    void )
```

This function handles Undefined instruction or illegal state.

## 9.34 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/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`

### 9.34.1 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) 2020-2024 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.

### 9.34.2 Function Documentation

#### 9.34.2.1 `__attribute__()`

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

#### 9.34.2.2 `__io_getchar()`

```
int __io_getchar (  
    void ) [extern]
```

#### 9.34.2.3 `__io_putchar()`

```
int __io_putchar (  
    int ch ) [extern]
```

#### 9.34.2.4 `_close()`

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

#### 9.34.2.5 `_execve()`

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

#### 9.34.2.6 `_exit()`

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

#### 9.34.2.7 `_fork()`

```
int _fork (
    void )
```

#### 9.34.2.8 `_fstat()`

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

#### 9.34.2.9 `_getpid()`

```
int _getpid (
    void )
```

#### 9.34.2.10 `_isatty()`

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

#### 9.34.2.11 `_kill()`

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

#### 9.34.2.12 `_link()`

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

#### 9.34.2.13 `_lseek()`

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

#### 9.34.2.14 `_open()`

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

#### 9.34.2.15 `_stat()`

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

#### 9.34.2.16 `_times()`

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

#### 9.34.2.17 `_unlink()`

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

#### 9.34.2.18 `_wait()`

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

#### 9.34.2.19 `initialise_monitor_handles()`

```
void initialise_monitor_handles ( )
```

### 9.34.3 Variable Documentation

#### 9.34.3.1 `environ`

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



## 9.35 Ball\_Launcher\_Controller/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`

### 9.35.1 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) 2020-2024 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.

## 9.35.2 Function Documentation

### 9.35.2.1 `__attribute__()`

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

### 9.35.2.2 `__io_getchar()`

```
int __io_getchar (  
    void ) [extern]
```

### 9.35.2.3 `__io_putchar()`

```
int __io_putchar (  
    int ch ) [extern]
```

### 9.35.2.4 `_close()`

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

### 9.35.2.5 `_execve()`

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

### 9.35.2.6 `_exit()`

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

### 9.35.2.7 `_fork()`

```
int _fork (  
    void )
```

### 9.35.2.8 `_fstat()`

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

### 9.35.2.9 `_getpid()`

```
int _getpid (
    void )
```

### 9.35.2.10 `_isatty()`

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

### 9.35.2.11 `_kill()`

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

### 9.35.2.12 `_link()`

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

### 9.35.2.13 `_lseek()`

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

### 9.35.2.14 `_open()`

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

### 9.35.2.15 `_stat()`

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

### 9.35.2.16 `_times()`

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

**9.35.2.17 \_unlink()**

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

**9.35.2.18 \_wait()**

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

**9.35.2.19 initialise\_monitor\_handles()**

```
void initialise_monitor_handles ( )
```

**9.35.3 Variable Documentation****9.35.3.1 environ**

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

## 9.36 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/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*

**9.36.1 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) 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.

## 9.36.2 Function Documentation

### 9.36.2.1 \_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

## 9.37 Ball\_Launcher\_Controller/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

### 9.37.1 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) 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.

**9.37.2 Function Documentation****9.37.2.1 `_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**

<code>incr</code>	Memory size
-------------------	-------------

**Returns**

Pointer to allocated memory

## 9.38 **Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/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}

### 9.38.1 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.

## 9.39 Ball\_Launcher\_Controller/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}

### 9.39.1 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.



## 9.40 pages/hardware.c File Reference

Documentation for Hardware.

### 9.40.1 Detailed Description

Documentation for Hardware.

## 9.41 pages/mainpage.c File Reference

Main Page Documentation for Project.

### 9.41.1 Detailed Description

Main Page Documentation for Project.

## 9.42 pages/software.c File Reference

Documentation for Software.

### 9.42.1 Detailed Description

Documentation for Software.



# Index

- `__attribute__`
    - `syscalls.c`, [150](#), [154](#)
    - `syscalls_C.c`, [58](#)
  - `__io_getchar`
    - `syscalls.c`, [150](#), [154](#)
    - `syscalls_C.c`, [58](#)
  - `__io_putchar`
    - `syscalls.c`, [150](#), [154](#)
    - `syscalls_C.c`, [58](#)
  - `_close`
    - `syscalls.c`, [150](#), [154](#)
    - `syscalls_C.c`, [58](#)
  - `_execve`
    - `syscalls.c`, [150](#), [154](#)
    - `syscalls_C.c`, [58](#)
  - `_exit`
    - `syscalls.c`, [151](#), [154](#)
    - `syscalls_C.c`, [58](#)
  - `_fork`
    - `syscalls.c`, [151](#), [154](#)
    - `syscalls_C.c`, [58](#)
  - `_fstat`
    - `syscalls.c`, [151](#), [154](#)
    - `syscalls_C.c`, [58](#)
  - `_getpid`
    - `syscalls.c`, [151](#), [154](#)
    - `syscalls_C.c`, [58](#)
  - `_isatty`
    - `syscalls.c`, [151](#), [155](#)
    - `syscalls_C.c`, [59](#)
  - `_kill`
    - `syscalls.c`, [151](#), [155](#)
    - `syscalls_C.c`, [59](#)
  - `_link`
    - `syscalls.c`, [151](#), [155](#)
    - `syscalls_C.c`, [59](#)
  - `_lseek`
    - `syscalls.c`, [151](#), [155](#)
    - `syscalls_C.c`, [59](#)
  - `_open`
    - `syscalls.c`, [152](#), [155](#)
    - `syscalls_C.c`, [59](#)
  - `_sbrk`
    - `system.c`, [157](#), [158](#)
    - `system_C.c`, [61](#)
  - `_stat`
    - `syscalls.c`, [152](#), [155](#)
    - `syscalls_C.c`, [59](#)
  - `_times`

- `syscalls.c`, [152](#), [155](#)
  - `syscalls_C.c`, [59](#)
- `_unlink`
  - `syscalls.c`, [152](#), [155](#)
  - `syscalls_C.c`, [59](#)
- `_wait`
  - `syscalls.c`, [152](#), [156](#)
  - `syscalls_C.c`, [60](#)

- `addr`
  - `MPU6050`, [25](#)
- `AHBPrescTable`
  - `STM32F4xx_System_Private_Variables`, [19](#)
- `APBPrescTable`
  - `STM32F4xx_System_Private_Variables`, [19](#)
- `assert_param`
  - `stm32f4xx_hal_conf.h`, [83](#), [97](#), [111](#)
  - `stm32f4xx_hal_conf_C.h`, [35](#)

- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/main.h`, [71](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/main_C.h`, [31](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/mpu6050.h`, [74](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/stm32f4xx_h`, [81](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/stm32f4xx_h`, [32](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/stm32f4xx_it`, [122](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Inc/stm32f4xx_it`, [46](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/main.c`, [129](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/main_C.c`, [48](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/mpu6050.c`, [135](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/stm32f4xx_h`, [140](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/stm32f4xx_h`, [52](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/stm32f4xx_i`, [145](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/stm32f4xx_i`, [54](#)
- `Ball_Launcher_Controller/Ball_Launcher_Controller/Core/Src/syscalls.c`, [149](#)

Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/Src/syscalls.c, 27  
 57  
 counter2  
 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/Src/system\_stm32f4xx.c, 27  
 156  
 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/Src/systemem\_C.c, 423  
 60  
 D4215\_init, 63  
 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/Src/system\_stm32f4xx.c, 421  
 158  
 D4215\_init  
 Ball\_Launcher\_Controller/Ball\_Launcher\_Controller/Core/Src/system\_stm32f4xx\_C.c, 421  
 61  
 D4215\_set  
 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/d4215.h, 421  
 62  
 D4215X, 23  
 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/led.h, channel, 23  
 64  
 speed, 23  
 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/main.h, timer, 23  
 72  
 DATA\_CACHE\_ENABLE  
 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/radio\_driver.h, stm32f4xx\_hal\_conf.h, 83, 97, 111  
 65  
 stm32f4xx\_hal\_conf\_C.h, 35  
 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/stepper\_driver.h, Debouncer\_Handler  
 67  
 stm32f4xx\_it.c, 145, 148  
 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/stm32f4xx\_hal\_conf.h, stm32f4xx\_it.h, 123, 125, 127  
 95  
 stm32f4xx\_it\_C.c, 55  
 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/stm32f4xx\_hal\_conf\_C.h, stm32f4xx\_it\_C.h, 47  
 124  
 Delay\_us  
 Ball\_Launcher\_Controller/Ball\_Launcher\_Main/Core/Inc/switch\_stepper\_driver.h, 68  
 70  
 DIR\_PIN  
 Ball\_Launcher\_Controller/Core/Inc/main.h, 73  
 StepperX, 28  
 Ball\_Launcher\_Controller/Core/Inc/mpu6050.h, 77  
 DP83848\_PHY\_ADDRESS  
 Ball\_Launcher\_Controller/Core/Inc/stm32f4xx\_hal\_conf.h, stm32f4xx\_hal\_conf.h, 84, 97, 111  
 108  
 stm32f4xx\_hal\_conf\_C.h, 35  
 Ball\_Launcher\_Controller/Core/Inc/stm32f4xx\_it.h, 126  
 dt  
 Ball\_Launcher\_Controller/Core/Src/main.c, 131  
 MPU6050, 25  
 Ball\_Launcher\_Controller/Core/Src/mpu6050.c, 137  
 EN\_PIN  
 Ball\_Launcher\_Controller/Core/Src/stm32f4xx\_hal\_msp.c, StepperX, 28  
 142  
 environ  
 Ball\_Launcher\_Controller/Core/Src/stm32f4xx\_it.c, 147  
 syscalls.c, 152, 156  
 Ball\_Launcher\_Controller/Core/Src/syscalls.c, 153  
 syscalls\_C.c, 60  
 Ball\_Launcher\_Controller/Core/Src/systemem.c, 157  
 Error\_Handler  
 Ball\_Launcher\_Controller/Core/Src/system\_stm32f4xx.c, main.c, 129, 132  
 160  
 main.h, 71–73  
 buffer  
 main\_C.c, 49  
 main\_C.h, 31  
 BusFault\_Handler  
 stm32f4xx\_it.c, 145, 148  
 stm32f4xx\_it.h, 123, 125, 127  
 stm32f4xx\_it\_C.c, 55  
 stm32f4xx\_it\_C.h, 47  
 capturePulseWidth  
 radio\_driver.h, 66  
 channel  
 D4215X, 23  
 channel1  
 RadioX, 27  
 channel2  
 RadioX, 27  
 CMSIS, 17  
 counter1

- GPIOx
  - LEDX, [24](#)
  - StepperX, [29](#)
  - SwitchX, [29](#)
- gX
  - main.c, [130](#), [134](#)
  - main\_C.c, [50](#)
  - MPU6050, [25](#)
- gX\_offset
  - MPU6050, [25](#)
- gY
  - main.c, [130](#), [134](#)
  - main\_C.c, [50](#)
  - MPU6050, [25](#)
- gY\_offset
  - MPU6050, [26](#)
- gZ
  - main.c, [130](#), [134](#)
  - main\_C.c, [50](#)
  - MPU6050, [26](#)
- gZ\_offset
  - MPU6050, [26](#)
- HAL\_CORTEX\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [84](#), [98](#), [111](#)
  - stm32f4xx\_hal\_conf\_C.h, [35](#)
- HAL\_DMA\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [84](#), [98](#), [112](#)
  - stm32f4xx\_hal\_conf\_C.h, [36](#)
- HAL\_EXTI\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [84](#), [98](#), [112](#)
  - stm32f4xx\_hal\_conf\_C.h, [36](#)
- HAL\_FLASH\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [85](#), [98](#), [112](#)
  - stm32f4xx\_hal\_conf\_C.h, [36](#)
- HAL\_GPIO\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [85](#), [98](#), [112](#)
  - stm32f4xx\_hal\_conf\_C.h, [36](#)
- HAL\_I2C\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [85](#), [112](#)
  - stm32f4xx\_hal\_conf\_C.h, [36](#)
- HAL\_I2C\_MspDeInit
  - stm32f4xx\_hal\_msp.c, [140](#), [143](#)
  - stm32f4xx\_hal\_msp\_C.c, [52](#)
- HAL\_I2C\_MspInit
  - stm32f4xx\_hal\_msp.c, [141](#), [143](#)
  - stm32f4xx\_hal\_msp\_C.c, [53](#)
- HAL\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [85](#), [98](#), [112](#)
  - stm32f4xx\_hal\_conf\_C.h, [36](#)
- HAL\_MspInit
  - stm32f4xx\_hal\_msp.c, [141](#), [144](#)
  - stm32f4xx\_hal\_msp\_C.c, [53](#)
- HAL\_PWR\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [85](#), [99](#), [112](#)
  - stm32f4xx\_hal\_conf\_C.h, [36](#)
- HAL\_RCC\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [85](#), [99](#), [112](#)
  - stm32f4xx\_hal\_conf\_C.h, [36](#)
- HAL\_TIM\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [99](#)
- HAL\_TIM\_MspPostInit
  - main.h, [72](#)
- HAL\_UART\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [85](#), [99](#), [112](#)
  - stm32f4xx\_hal\_conf\_C.h, [36](#)
- HAL\_UART\_MspDeInit
  - stm32f4xx\_hal\_msp.c, [141](#), [144](#)
  - stm32f4xx\_hal\_msp\_C.c, [53](#)
- HAL\_UART\_MspInit
  - stm32f4xx\_hal\_msp.c, [142](#), [144](#)
  - stm32f4xx\_hal\_msp\_C.c, [54](#)
- HardFault\_Handler
  - stm32f4xx\_it.c, [146](#), [148](#)
  - stm32f4xx\_it.h, [123](#), [125](#), [127](#)
  - stm32f4xx\_it\_C.c, [55](#)
  - stm32f4xx\_it\_C.h, [47](#)
- Hardware Documentation, [5](#)
- hi2c
  - MPU6050, [26](#)
- hi2c1
  - main.c, [130](#), [134](#)
  - main\_C.c, [50](#)
- HSE\_STARTUP\_TIMEOUT
  - stm32f4xx\_hal\_conf.h, [85](#), [99](#), [112](#)
  - stm32f4xx\_hal\_conf\_C.h, [36](#)
- HSE\_VALUE
  - stm32f4xx\_hal\_conf.h, [85](#), [99](#), [113](#)
  - stm32f4xx\_hal\_conf\_C.h, [37](#)
  - STM32F4xx\_System\_Private\_Includes, [18](#)
- HSI\_VALUE
  - stm32f4xx\_hal\_conf.h, [86](#), [99](#), [113](#)
  - stm32f4xx\_hal\_conf\_C.h, [37](#)
  - STM32F4xx\_System\_Private\_Includes, [18](#)
- huart1
  - main.c, [131](#), [134](#)
  - main\_C.c, [50](#)
- huart2
  - main.c, [131](#), [134](#)
  - main\_C.c, [51](#)
- initialise\_monitor\_handles
  - syscalls.c, [152](#), [156](#)
  - syscalls\_C.c, [60](#)
- INSTRUCTION\_CACHE\_ENABLE
  - stm32f4xx\_hal\_conf.h, [86](#), [99](#), [113](#)
  - stm32f4xx\_hal\_conf\_C.h, [37](#)
- led.h
  - LED\_init, [64](#)
  - LED\_off, [65](#)
  - LED\_on, [65](#)
  - LED\_toggle, [65](#)
- LED\_init
  - led.h, [64](#)
- LED\_off
  - led.h, [65](#)
- LED\_on

- led.h, 65
- LED\_toggle
  - led.h, 65
- LEDX, 24
  - GPIOx, 24
  - PIN, 24
- LSE\_STARTUP\_TIMEOUT
  - stm32f4xx\_hal\_conf.h, 86, 100, 113
  - stm32f4xx\_hal\_conf\_C.h, 37
- LSE\_VALUE
  - stm32f4xx\_hal\_conf.h, 86, 100, 113
  - stm32f4xx\_hal\_conf\_C.h, 37
- LSI\_VALUE
  - stm32f4xx\_hal\_conf.h, 86, 100, 113
  - stm32f4xx\_hal\_conf\_C.h, 37
- MAC\_ADDR0
  - stm32f4xx\_hal\_conf.h, 86, 100, 114
  - stm32f4xx\_hal\_conf\_C.h, 38
- MAC\_ADDR1
  - stm32f4xx\_hal\_conf.h, 87, 100, 114
  - stm32f4xx\_hal\_conf\_C.h, 38
- MAC\_ADDR2
  - stm32f4xx\_hal\_conf.h, 87, 100, 114
  - stm32f4xx\_hal\_conf\_C.h, 38
- MAC\_ADDR3
  - stm32f4xx\_hal\_conf.h, 87, 100, 114
  - stm32f4xx\_hal\_conf\_C.h, 38
- MAC\_ADDR4
  - stm32f4xx\_hal\_conf.h, 87, 100, 114
  - stm32f4xx\_hal\_conf\_C.h, 38
- MAC\_ADDR5
  - stm32f4xx\_hal\_conf.h, 87, 101, 114
  - stm32f4xx\_hal\_conf\_C.h, 38
- main
  - main.c, 130, 132
  - main\_C.c, 49
- MAIN\_PAGE, 1
- main.c
  - buffer, 130, 134
  - Error\_Handler, 129, 132
  - gX, 130, 134
  - gY, 130, 134
  - gZ, 130, 134
  - hi2c1, 130, 134
  - huart1, 131, 134
  - huart2, 131, 134
  - main, 130, 132
  - move, 131, 134
  - shot, 131, 134
  - stat, 131, 135
  - SystemClock\_Config, 130, 132
- main.h
  - Error\_Handler, 71–73
  - HAL\_TIM\_MspPostInit, 72
- main\_C.c
  - buffer, 50
  - Error\_Handler, 49
  - gX, 50
  - gY, 50
  - gZ, 50
  - hi2c1, 50
  - huart1, 50
  - huart2, 51
  - main, 49
  - move, 51
  - shot, 51
  - stat, 51
  - STATE, 51
  - STATE\_0\_INIT, 51
  - STATE\_1\_ERROR, 51
  - STATE\_2\_IMU, 51
  - STATE\_3\_BUTTON\_LED, 51
  - STATE\_4\_TRANSFER, 51
  - SystemClock\_Config, 50
- main\_C.h
  - Error\_Handler, 31
- MemManage\_Handler
  - stm32f4xx\_it.c, 146, 148
  - stm32f4xx\_it.h, 123, 125, 128
  - stm32f4xx\_it\_C.c, 55
  - stm32f4xx\_it\_C.h, 47
- move
  - main.c, 131, 134
  - main\_C.c, 51
- MPU6050, 25
  - addr, 25
  - dt, 25
  - gX, 25
  - gX\_offset, 25
  - gY, 25
  - gY\_offset, 26
  - gZ, 26
  - gZ\_offset, 26
  - hi2c, 26
  - status, 26
- mpu6050.c
  - mpu6050\_calibrate, 135, 138
  - mpu6050\_get\_gX, 136, 138
  - mpu6050\_get\_gY, 136, 138
  - mpu6050\_get\_gZ, 136, 139
  - mpu6050\_init, 137, 139
  - mpu6050\_update, 137, 139
- mpu6050.h
  - mpu6050\_calibrate, 74, 78
  - mpu6050\_get\_gX, 75, 78
  - mpu6050\_get\_gY, 75, 79
  - mpu6050\_get\_gZ, 75, 79
  - mpu6050\_get\_X, 76, 79
  - mpu6050\_get\_Y, 76, 80
  - mpu6050\_get\_Z, 76, 80
  - mpu6050\_init, 77, 80
  - mpu6050\_update, 77, 81
- mpu6050\_calibrate
  - mpu6050.c, 135, 138
  - mpu6050.h, 74, 78
- mpu6050\_get\_gX

- mpu6050.c, [136](#), [138](#)
- mpu6050.h, [75](#), [78](#)
- mpu6050\_get\_gY
  - mpu6050.c, [136](#), [138](#)
  - mpu6050.h, [75](#), [79](#)
- mpu6050\_get\_gZ
  - mpu6050.c, [136](#), [139](#)
  - mpu6050.h, [75](#), [79](#)
- mpu6050\_get\_X
  - mpu6050.h, [76](#), [79](#)
- mpu6050\_get\_Y
  - mpu6050.h, [76](#), [80](#)
- mpu6050\_get\_Z
  - mpu6050.h, [76](#), [80](#)
- mpu6050\_init
  - mpu6050.c, [137](#), [139](#)
  - mpu6050.h, [77](#), [80](#)
- mpu6050\_update
  - mpu6050.c, [137](#), [139](#)
  - mpu6050.h, [77](#), [81](#)
- NMI\_Handler
  - stm32f4xx\_it.c, [146](#), [148](#)
  - stm32f4xx\_it.h, [123](#), [125](#), [128](#)
  - stm32f4xx\_it\_C.c, [56](#)
  - stm32f4xx\_it\_C.h, [47](#)
- pages/hardware.c, [161](#)
- pages/mainpage.c, [161](#)
- pages/software.c, [161](#)
- PendSV\_Handler
  - stm32f4xx\_it.c, [146](#), [148](#)
  - stm32f4xx\_it.h, [123](#), [125](#), [128](#)
  - stm32f4xx\_it\_C.c, [56](#)
  - stm32f4xx\_it\_C.h, [47](#)
- PHY\_AUTONEGO\_COMPLETE
  - stm32f4xx\_hal\_conf.h, [87](#), [101](#), [114](#)
  - stm32f4xx\_hal\_conf\_C.h, [38](#)
- PHY\_AUTONEGOTIATION
  - stm32f4xx\_hal\_conf.h, [87](#), [101](#), [114](#)
  - stm32f4xx\_hal\_conf\_C.h, [38](#)
- PHY\_BCR
  - stm32f4xx\_hal\_conf.h, [87](#), [101](#), [115](#)
  - stm32f4xx\_hal\_conf\_C.h, [39](#)
- PHY\_BSR
  - stm32f4xx\_hal\_conf.h, [87](#), [101](#), [115](#)
  - stm32f4xx\_hal\_conf\_C.h, [39](#)
- PHY\_CONFIG\_DELAY
  - stm32f4xx\_hal\_conf.h, [88](#), [101](#), [115](#)
  - stm32f4xx\_hal\_conf\_C.h, [39](#)
- PHY\_DUPLEX\_STATUS
  - stm32f4xx\_hal\_conf.h, [88](#), [101](#), [115](#)
  - stm32f4xx\_hal\_conf\_C.h, [39](#)
- PHY\_FULLDUPLEX\_100M
  - stm32f4xx\_hal\_conf.h, [88](#), [101](#), [115](#)
  - stm32f4xx\_hal\_conf\_C.h, [39](#)
- PHY\_FULLDUPLEX\_10M
  - stm32f4xx\_hal\_conf.h, [88](#), [102](#), [115](#)
  - stm32f4xx\_hal\_conf\_C.h, [39](#)
- PHY\_HALFDUPLEX\_100M
  - stm32f4xx\_hal\_conf.h, [88](#), [102](#), [115](#)
  - stm32f4xx\_hal\_conf\_C.h, [39](#)
- PHY\_HALFDUPLEX\_10M
  - stm32f4xx\_hal\_conf.h, [88](#), [102](#), [116](#)
  - stm32f4xx\_hal\_conf\_C.h, [40](#)
- PHY\_ISOLATE
  - stm32f4xx\_hal\_conf.h, [88](#), [102](#), [116](#)
  - stm32f4xx\_hal\_conf\_C.h, [40](#)
- PHY\_JABBER\_DETECTION
  - stm32f4xx\_hal\_conf.h, [89](#), [102](#), [116](#)
  - stm32f4xx\_hal\_conf\_C.h, [40](#)
- PHY\_LINKED\_STATUS
  - stm32f4xx\_hal\_conf.h, [89](#), [102](#), [116](#)
  - stm32f4xx\_hal\_conf\_C.h, [40](#)
- PHY\_LOOPBACK
  - stm32f4xx\_hal\_conf.h, [89](#), [102](#), [116](#)
  - stm32f4xx\_hal\_conf\_C.h, [40](#)
- PHY\_POWERDOWN
  - stm32f4xx\_hal\_conf.h, [89](#), [103](#), [116](#)
  - stm32f4xx\_hal\_conf\_C.h, [40](#)
- PHY\_READ\_TO
  - stm32f4xx\_hal\_conf.h, [89](#), [103](#), [116](#)
  - stm32f4xx\_hal\_conf\_C.h, [40](#)
- PHY\_RESET
  - stm32f4xx\_hal\_conf.h, [89](#), [103](#), [117](#)
  - stm32f4xx\_hal\_conf\_C.h, [41](#)
- PHY\_RESET\_DELAY
  - stm32f4xx\_hal\_conf.h, [89](#), [103](#), [117](#)
  - stm32f4xx\_hal\_conf\_C.h, [41](#)
- PHY\_RESTART\_AUTONEGOTIATION
  - stm32f4xx\_hal\_conf.h, [90](#), [103](#), [117](#)
  - stm32f4xx\_hal\_conf\_C.h, [41](#)
- PHY\_SPEED\_STATUS
  - stm32f4xx\_hal\_conf.h, [90](#), [103](#), [117](#)
  - stm32f4xx\_hal\_conf\_C.h, [41](#)
- PHY\_SR
  - stm32f4xx\_hal\_conf.h, [90](#), [103](#), [117](#)
  - stm32f4xx\_hal\_conf\_C.h, [41](#)
- PHY\_WRITE\_TO
  - stm32f4xx\_hal\_conf.h, [90](#), [103](#), [117](#)
  - stm32f4xx\_hal\_conf\_C.h, [41](#)
- PIN
  - LEDX, [24](#)
  - SwitchX, [29](#)
- PREFETCH\_ENABLE
  - stm32f4xx\_hal\_conf.h, [90](#), [104](#), [117](#)
  - stm32f4xx\_hal\_conf\_C.h, [41](#)
- pulseWidth
  - RadioX, [27](#)
- radio\_driver.h
  - capturePulseWidth, [66](#)
  - Radio\_getPulseWidth, [66](#)
  - Radio\_init, [66](#)
  - update, [67](#)
- Radio\_getPulseWidth
  - radio\_driver.h, [66](#)
- Radio\_init

- radio\_driver.h, 66
- RadioX, 26
  - channel1, 27
  - channel2, 27
  - counter1, 27
  - counter2, 27
  - pulseWidth, 27
  - sum1, 27
  - sum2, 28
  - timer, 28
- shot
  - main.c, 131, 134
  - main\_C.c, 51
- Software Documentation, 9
- speed
  - D4215X, 23
- stat
  - main.c, 131, 135
  - main\_C.c, 51
- STATE
  - main\_C.c, 51
- STATE\_0\_INIT
  - main\_C.c, 51
- STATE\_1\_ERROR
  - main\_C.c, 51
- STATE\_2\_IMU
  - main\_C.c, 51
- STATE\_3\_BUTTON\_LED
  - main\_C.c, 51
- STATE\_4\_TRANSFER
  - main\_C.c, 51
- status
  - MPU6050, 26
  - SwitchX, 30
- Stepper\_disable
  - stepper\_driver.h, 68
- stepper\_driver.h
  - Delay\_us, 68
  - Stepper\_disable, 68
  - Stepper\_enable, 69
  - Stepper\_init, 69
  - Stepper\_setspeed, 69
  - SysTick\_Init, 69
- Stepper\_enable
  - stepper\_driver.h, 69
- Stepper\_init
  - stepper\_driver.h, 69
- Stepper\_setspeed
  - stepper\_driver.h, 69
- StepperX, 28
  - DIR\_PIN, 28
  - EN\_PIN, 28
  - GPIOx, 29
  - STP\_PIN, 29
- stm32f4xx\_hal\_conf.h
  - assert\_param, 83, 97, 111
  - DATA\_CACHE\_ENABLE, 83, 97, 111
  - DP83848\_PHY\_ADDRESS, 84, 97, 111
  - ETH\_RX\_BUF\_SIZE, 84, 97, 111
  - ETH\_RXBUFNB, 84, 97, 111
  - ETH\_TX\_BUF\_SIZE, 84, 98, 111
  - ETH\_TXBUFNB, 84, 98, 111
  - EXTERNAL\_CLOCK\_VALUE, 84, 98, 111
  - HAL\_CORTEX\_MODULE\_ENABLED, 84, 98, 111
  - HAL\_DMA\_MODULE\_ENABLED, 84, 98, 112
  - HAL\_EXTI\_MODULE\_ENABLED, 84, 98, 112
  - HAL\_FLASH\_MODULE\_ENABLED, 85, 98, 112
  - HAL\_GPIO\_MODULE\_ENABLED, 85, 98, 112
  - HAL\_I2C\_MODULE\_ENABLED, 85, 112
  - HAL\_MODULE\_ENABLED, 85, 98, 112
  - HAL\_PWR\_MODULE\_ENABLED, 85, 99, 112
  - HAL\_RCC\_MODULE\_ENABLED, 85, 99, 112
  - HAL\_TIM\_MODULE\_ENABLED, 99
  - HAL\_UART\_MODULE\_ENABLED, 85, 99, 112
  - HSE\_STARTUP\_TIMEOUT, 85, 99, 112
  - HSE\_VALUE, 85, 99, 113
  - HSI\_VALUE, 86, 99, 113
  - INSTRUCTION\_CACHE\_ENABLE, 86, 99, 113
  - LSE\_STARTUP\_TIMEOUT, 86, 100, 113
  - LSE\_VALUE, 86, 100, 113
  - LSI\_VALUE, 86, 100, 113
  - MAC\_ADDR0, 86, 100, 114
  - MAC\_ADDR1, 87, 100, 114
  - MAC\_ADDR2, 87, 100, 114
  - MAC\_ADDR3, 87, 100, 114
  - MAC\_ADDR4, 87, 100, 114
  - MAC\_ADDR5, 87, 101, 114
  - PHY\_AUTONEGO\_COMPLETE, 87, 101, 114
  - PHY\_AUTONEGOTIATION, 87, 101, 114
  - PHY\_BCR, 87, 101, 115
  - PHY\_BSR, 87, 101, 115
  - PHY\_CONFIG\_DELAY, 88, 101, 115
  - PHY\_DUPLEX\_STATUS, 88, 101, 115
  - PHY\_FULLDUPLEX\_100M, 88, 101, 115
  - PHY\_FULLDUPLEX\_10M, 88, 102, 115
  - PHY\_HALFDUPLEX\_100M, 88, 102, 115
  - PHY\_HALFDUPLEX\_10M, 88, 102, 116
  - PHY\_ISOLATE, 88, 102, 116
  - PHY\_JABBER\_DETECTION, 89, 102, 116
  - PHY\_LINKED\_STATUS, 89, 102, 116
  - PHY\_LOOPBACK, 89, 102, 116
  - PHY\_POWERDOWN, 89, 103, 116
  - PHY\_READ\_TO, 89, 103, 116
  - PHY\_RESET, 89, 103, 117
  - PHY\_RESET\_DELAY, 89, 103, 117
  - PHY\_RESTART\_AUTONEGOTIATION, 90, 103, 117
  - PHY\_SPEED\_STATUS, 90, 103, 117
  - PHY\_SR, 90, 103, 117
  - PHY\_WRITE\_TO, 90, 103, 117
  - PREFETCH\_ENABLE, 90, 104, 117
  - TICK\_INT\_PRIORITY, 90, 104, 117
  - USE\_HAL\_ADC\_REGISTER\_CALLBACKS, 90, 104, 118
  - USE\_HAL\_CAN\_REGISTER\_CALLBACKS, 90, 104, 118



- USE\_HAL\_CEC\_REGISTER\_CALLBACKS, [91](#),  
[104](#), [118](#)
- USE\_HAL\_Cryp\_REGISTER\_CALLBACKS, [91](#),  
[104](#), [118](#)
- USE\_HAL\_DAC\_REGISTER\_CALLBACKS, [91](#),  
[104](#), [118](#)
- USE\_HAL\_DCMI\_REGISTER\_CALLBACKS, [91](#),  
[104](#), [118](#)
- USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS,  
[91](#), [104](#), [118](#)
- USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS,  
[91](#), [104](#), [118](#)
- USE\_HAL\_DSI\_REGISTER\_CALLBACKS, [91](#),  
[105](#), [118](#)
- USE\_HAL\_ETH\_REGISTER\_CALLBACKS, [91](#),  
[105](#), [118](#)
- USE\_HAL\_FMPI2C\_REGISTER\_CALLBACKS,  
[91](#), [105](#), [119](#)
- USE\_HAL\_FMPMBUS\_REGISTER\_CALLBACKS,  
[91](#), [105](#), [119](#)
- USE\_HAL\_HASH\_REGISTER\_CALLBACKS, [92](#),  
[105](#), [119](#)
- USE\_HAL\_HCD\_REGISTER\_CALLBACKS, [92](#),  
[105](#), [119](#)
- USE\_HAL\_I2C\_REGISTER\_CALLBACKS, [92](#),  
[105](#), [119](#)
- USE\_HAL\_I2S\_REGISTER\_CALLBACKS, [92](#),  
[105](#), [119](#)
- USE\_HAL\_IRDA\_REGISTER\_CALLBACKS, [92](#),  
[105](#), [119](#)
- USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS, [92](#),  
[105](#), [119](#)
- USE\_HAL\_LTDC\_REGISTER\_CALLBACKS, [92](#),  
[106](#), [119](#)
- USE\_HAL\_MMC\_REGISTER\_CALLBACKS, [92](#),  
[106](#), [119](#)
- USE\_HAL\_NAND\_REGISTER\_CALLBACKS, [92](#),  
[106](#), [120](#)
- USE\_HAL\_NOR\_REGISTER\_CALLBACKS, [92](#),  
[106](#), [120](#)
- USE\_HAL\_PCCARD\_REGISTER\_CALLBACKS,  
[93](#), [106](#), [120](#)
- USE\_HAL\_PCD\_REGISTER\_CALLBACKS, [93](#),  
[106](#), [120](#)
- USE\_HAL\_QSPI\_REGISTER\_CALLBACKS, [93](#),  
[106](#), [120](#)
- USE\_HAL\_RNG\_REGISTER\_CALLBACKS, [93](#),  
[106](#), [120](#)
- USE\_HAL\_RTC\_REGISTER\_CALLBACKS, [93](#),  
[106](#), [120](#)
- USE\_HAL\_SAI\_REGISTER\_CALLBACKS, [93](#),  
[106](#), [120](#)
- USE\_HAL\_SD\_REGISTER\_CALLBACKS, [93](#),  
[107](#), [120](#)
- USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS,  
[93](#), [107](#), [120](#)
- USE\_HAL\_SMARTCARD\_REGISTER\_CALLBACKS,  
[93](#), [107](#), [121](#)
- USE\_HAL\_SMBUS\_REGISTER\_CALLBACKS,  
[93](#), [107](#), [121](#)
- USE\_HAL\_SPDIFRX\_REGISTER\_CALLBACKS,  
[94](#), [107](#), [121](#)
- USE\_HAL\_SPI\_REGISTER\_CALLBACKS, [94](#),  
[107](#), [121](#)
- USE\_HAL\_SRAM\_REGISTER\_CALLBACKS, [94](#),  
[107](#), [121](#)
- USE\_HAL\_TIM\_REGISTER\_CALLBACKS, [94](#),  
[107](#), [121](#)
- USE\_HAL\_UART\_REGISTER\_CALLBACKS, [94](#),  
[107](#), [121](#)
- USE\_HAL\_USART\_REGISTER\_CALLBACKS, [94](#),  
[107](#), [121](#)
- USE\_HAL\_WWDG\_REGISTER\_CALLBACKS, [94](#),  
[108](#), [121](#)
- USE\_RTOS, [94](#), [108](#), [121](#)
- USE\_SPI\_CRC, [94](#), [108](#), [122](#)
- VDD\_VALUE, [94](#), [108](#), [122](#)
- stm32f4xx\_hal\_conf\_C.h
  - assert\_param, [35](#)
  - DATA\_CACHE\_ENABLE, [35](#)
  - DP83848\_PHY\_ADDRESS, [35](#)
  - ETH\_RX\_BUF\_SIZE, [35](#)
  - ETH\_RXBUFNB, [35](#)
  - ETH\_TX\_BUF\_SIZE, [35](#)
  - ETH\_TXBUFNB, [35](#)
  - EXTERNAL\_CLOCK\_VALUE, [35](#)
  - HAL\_CORTEX\_MODULE\_ENABLED, [35](#)
  - HAL\_DMA\_MODULE\_ENABLED, [36](#)
  - HAL\_EXTI\_MODULE\_ENABLED, [36](#)
  - HAL\_FLASH\_MODULE\_ENABLED, [36](#)
  - HAL\_GPIO\_MODULE\_ENABLED, [36](#)
  - HAL\_I2C\_MODULE\_ENABLED, [36](#)
  - HAL\_MODULE\_ENABLED, [36](#)
  - HAL\_PWR\_MODULE\_ENABLED, [36](#)
  - HAL\_RCC\_MODULE\_ENABLED, [36](#)
  - HAL\_UART\_MODULE\_ENABLED, [36](#)
  - HSE\_STARTUP\_TIMEOUT, [36](#)
  - HSE\_VALUE, [37](#)
  - HSI\_VALUE, [37](#)
  - INSTRUCTION\_CACHE\_ENABLE, [37](#)
  - LSE\_STARTUP\_TIMEOUT, [37](#)
  - LSE\_VALUE, [37](#)
  - LSI\_VALUE, [37](#)
  - MAC\_ADDR0, [38](#)
  - MAC\_ADDR1, [38](#)
  - MAC\_ADDR2, [38](#)
  - MAC\_ADDR3, [38](#)
  - MAC\_ADDR4, [38](#)
  - MAC\_ADDR5, [38](#)
  - PHY\_AUTONEGO\_COMPLETE, [38](#)
  - PHY\_AUTONEGOTIATION, [38](#)
  - PHY\_BCR, [39](#)
  - PHY\_BSR, [39](#)
  - PHY\_CONFIG\_DELAY, [39](#)
  - PHY\_DUPLEX\_STATUS, [39](#)
  - PHY\_FULLDUPLEX\_100M, [39](#)

- PHY\_FULLDUPLEX\_10M, [39](#)
- PHY\_HALFDUPLEX\_100M, [39](#)
- PHY\_HALFDUPLEX\_10M, [40](#)
- PHY\_ISOLATE, [40](#)
- PHY\_JABBER\_DETECTION, [40](#)
- PHY\_LINKED\_STATUS, [40](#)
- PHY\_LOOPBACK, [40](#)
- PHY\_POWERDOWN, [40](#)
- PHY\_READ\_TO, [40](#)
- PHY\_RESET, [41](#)
- PHY\_RESET\_DELAY, [41](#)
- PHY\_RESTART\_AUTONEGOTIATION, [41](#)
- PHY\_SPEED\_STATUS, [41](#)
- PHY\_SR, [41](#)
- PHY\_WRITE\_TO, [41](#)
- PREFETCH\_ENABLE, [41](#)
- TICK\_INT\_PRIORITY, [41](#)
- USE\_HAL\_ADC\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_CAN\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_CEC\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_CRYPT\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_DAC\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_DCMI\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_DSI\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_ETH\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_FMPI2C\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_FMPSMBUS\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_HASH\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_HCD\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_I2C\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_I2S\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_IRDA\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_LTDC\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_MMC\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_NAND\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_NOR\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_PCCARD\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_PCD\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_QSPI\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_RNG\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_RTC\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_SAI\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_SD\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_SMARTCARD\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_SMBUS\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_SPDIFRX\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_SPI\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_SRAM\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_TIM\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_UART\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_USART\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_WWDG\_REGISTER\_CALLBACKS, [45](#)
- USE\_RTOS, [45](#)
- USE\_SPI\_CRC, [46](#)
- VDD\_VALUE, [46](#)
- stm32f4xx\_hal\_msp.c
  - HAL\_I2C\_MspDeInit, [140](#), [143](#)
  - HAL\_I2C\_MspInit, [141](#), [143](#)
  - HAL\_MspInit, [141](#), [144](#)
  - HAL\_UART\_MspDeInit, [141](#), [144](#)
  - HAL\_UART\_MspInit, [142](#), [144](#)
- stm32f4xx\_hal\_msp\_C.c
  - HAL\_I2C\_MspDeInit, [52](#)
  - HAL\_I2C\_MspInit, [53](#)
  - HAL\_MspInit, [53](#)
  - HAL\_UART\_MspDeInit, [53](#)
  - HAL\_UART\_MspInit, [54](#)
- stm32f4xx\_it.c
  - BusFault\_Handler, [145](#), [148](#)
  - DebugMon\_Handler, [145](#), [148](#)
  - HardFault\_Handler, [146](#), [148](#)
  - MemManage\_Handler, [146](#), [148](#)
  - NMI\_Handler, [146](#), [148](#)
  - PendSV\_Handler, [146](#), [148](#)
  - SVC\_Handler, [146](#), [148](#)
  - SysTick\_Handler, [146](#), [149](#)
  - UsageFault\_Handler, [146](#), [149](#)
- stm32f4xx\_it.h
  - BusFault\_Handler, [123](#), [125](#), [127](#)
  - DebugMon\_Handler, [123](#), [125](#), [127](#)
  - HardFault\_Handler, [123](#), [125](#), [127](#)
  - MemManage\_Handler, [123](#), [125](#), [128](#)
  - NMI\_Handler, [123](#), [125](#), [128](#)
  - PendSV\_Handler, [123](#), [125](#), [128](#)
  - SVC\_Handler, [124](#), [126](#), [128](#)
  - SysTick\_Handler, [124](#), [126](#), [128](#)
  - TIM4\_IRQHandler, [126](#)
  - UsageFault\_Handler, [124](#), [126](#), [128](#)
  - USART1\_IRQHandler, [126](#)
- stm32f4xx\_it\_C.c
  - BusFault\_Handler, [55](#)
  - DebugMon\_Handler, [55](#)
  - HardFault\_Handler, [55](#)
  - MemManage\_Handler, [55](#)
  - NMI\_Handler, [56](#)
  - PendSV\_Handler, [56](#)
  - SVC\_Handler, [56](#)
  - SysTick\_Handler, [56](#)
  - UsageFault\_Handler, [56](#)
- stm32f4xx\_it\_C.h
  - BusFault\_Handler, [47](#)
  - DebugMon\_Handler, [47](#)
  - HardFault\_Handler, [47](#)
  - MemManage\_Handler, [47](#)
  - NMI\_Handler, [47](#)
  - PendSV\_Handler, [47](#)
  - SVC\_Handler, [48](#)
  - SysTick\_Handler, [48](#)

- UsageFault\_Handler, 48
- Stm32f4xx\_system, 17
- STM32F4xx\_System\_Private\_Defines, 19
- STM32F4xx\_System\_Private\_FunctionPrototypes, 20
- STM32F4xx\_System\_Private\_Functions, 20
  - SystemCoreClockUpdate, 20
  - SystemInit, 21
- STM32F4xx\_System\_Private\_Includes, 17
  - HSE\_VALUE, 18
  - HSI\_VALUE, 18
- STM32F4xx\_System\_Private\_Macros, 19
- STM32F4xx\_System\_Private\_TypesDefinitions, 19
- STM32F4xx\_System\_Private\_Variables, 19
  - AHBPrescTable, 19
  - APBPrescTable, 19
  - SystemCoreClock, 20
- STP\_PIN
  - StepperX, 29
- sum1
  - RadioX, 27
- sum2
  - RadioX, 28
- SVC\_Handler
  - stm32f4xx\_it.c, 146, 148
  - stm32f4xx\_it.h, 124, 126, 128
  - stm32f4xx\_it\_C.c, 56
  - stm32f4xx\_it\_C.h, 48
- switch.h
  - Switch\_getStatus, 70
  - Switch\_init, 71
- Switch\_getStatus
  - switch.h, 70
- Switch\_init
  - switch.h, 71
- SwitchX, 29
  - GPIOx, 29
  - PIN, 29
  - status, 30
- syscalls.c
  - \_\_attribute\_\_, 150, 154
  - \_\_io\_getchar, 150, 154
  - \_\_io\_putchar, 150, 154
  - \_close, 150, 154
  - \_execve, 150, 154
  - \_exit, 151, 154
  - \_fork, 151, 154
  - \_fstat, 151, 154
  - \_getpid, 151, 154
  - \_isatty, 151, 155
  - \_kill, 151, 155
  - \_link, 151, 155
  - \_lseek, 151, 155
  - \_open, 152, 155
  - \_stat, 152, 155
  - \_times, 152, 155
  - \_unlink, 152, 155
  - \_wait, 152, 156
  - environ, 152, 156
  - initialise\_monitor\_handles, 152, 156
  - syscalls\_C.c
    - \_\_attribute\_\_, 58
    - \_\_io\_getchar, 58
    - \_\_io\_putchar, 58
    - \_close, 58
    - \_execve, 58
    - \_exit, 58
    - \_fork, 58
    - \_fstat, 58
    - \_getpid, 58
    - \_isatty, 59
    - \_kill, 59
    - \_link, 59
    - \_lseek, 59
    - \_open, 59
    - \_stat, 59
    - \_times, 59
    - \_unlink, 59
    - \_wait, 60
    - environ, 60
    - initialise\_monitor\_handles, 60
- systemem.c
  - \_sbrk, 157, 158
- systemem\_C.c
  - \_sbrk, 61
- SystemClock\_Config
  - main.c, 130, 132
  - main\_C.c, 50
- SystemCoreClock
  - STM32F4xx\_System\_Private\_Variables, 20
- SystemCoreClockUpdate
  - STM32F4xx\_System\_Private\_Functions, 20
- SystemInit
  - STM32F4xx\_System\_Private\_Functions, 21
- SysTick\_Handler
  - stm32f4xx\_it.c, 146, 149
  - stm32f4xx\_it.h, 124, 126, 128
  - stm32f4xx\_it\_C.c, 56
  - stm32f4xx\_it\_C.h, 48
- SysTick\_Init
  - stepper\_driver.h, 69
- TICK\_INT\_PRIORITY
  - stm32f4xx\_hal\_conf.h, 90, 104, 117
  - stm32f4xx\_hal\_conf\_C.h, 41
- TIM4\_IRQHandler
  - stm32f4xx\_it.h, 126
- timer
  - D4215X, 23
  - RadioX, 28
- update
  - radio\_driver.h, 67
- UsageFault\_Handler
  - stm32f4xx\_it.c, 146, 149
  - stm32f4xx\_it.h, 124, 126, 128
  - stm32f4xx\_it\_C.c, 56
  - stm32f4xx\_it\_C.h, 48

USART1\_IRQHandler  
  stm32f4xx\_it.h, 126

USE\_HAL\_ADC\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 90, 104, 118  
  stm32f4xx\_hal\_conf\_C.h, 42

USE\_HAL\_CAN\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 90, 104, 118  
  stm32f4xx\_hal\_conf\_C.h, 42

USE\_HAL\_CEC\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 91, 104, 118  
  stm32f4xx\_hal\_conf\_C.h, 42

USE\_HAL\_Cryp\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 91, 104, 118  
  stm32f4xx\_hal\_conf\_C.h, 42

USE\_HAL\_DAC\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 91, 104, 118  
  stm32f4xx\_hal\_conf\_C.h, 42

USE\_HAL\_DCMI\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 91, 104, 118  
  stm32f4xx\_hal\_conf\_C.h, 42

USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 91, 104, 118  
  stm32f4xx\_hal\_conf\_C.h, 42

USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 91, 104, 118  
  stm32f4xx\_hal\_conf\_C.h, 42

USE\_HAL\_DSI\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 91, 105, 118  
  stm32f4xx\_hal\_conf\_C.h, 42

USE\_HAL\_ETH\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 91, 105, 118  
  stm32f4xx\_hal\_conf\_C.h, 42

USE\_HAL\_FMPI2C\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 91, 105, 119  
  stm32f4xx\_hal\_conf\_C.h, 43

USE\_HAL\_FMPSPMBUS\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 91, 105, 119  
  stm32f4xx\_hal\_conf\_C.h, 43

USE\_HAL\_HASH\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 92, 105, 119  
  stm32f4xx\_hal\_conf\_C.h, 43

USE\_HAL\_HCD\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 92, 105, 119  
  stm32f4xx\_hal\_conf\_C.h, 43

USE\_HAL\_I2C\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 92, 105, 119  
  stm32f4xx\_hal\_conf\_C.h, 43

USE\_HAL\_I2S\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 92, 105, 119  
  stm32f4xx\_hal\_conf\_C.h, 43

USE\_HAL\_IRDA\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 92, 105, 119  
  stm32f4xx\_hal\_conf\_C.h, 43

USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 92, 105, 119  
  stm32f4xx\_hal\_conf\_C.h, 43

USE\_HAL\_LTDC\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 92, 106, 119  
  stm32f4xx\_hal\_conf\_C.h, 43

USE\_HAL\_MMC\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 92, 106, 119  
  stm32f4xx\_hal\_conf\_C.h, 43

USE\_HAL\_NAND\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 92, 106, 120  
  stm32f4xx\_hal\_conf\_C.h, 44

USE\_HAL\_NOR\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 92, 106, 120  
  stm32f4xx\_hal\_conf\_C.h, 44

USE\_HAL\_PCCARD\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 93, 106, 120  
  stm32f4xx\_hal\_conf\_C.h, 44

USE\_HAL\_PCD\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 93, 106, 120  
  stm32f4xx\_hal\_conf\_C.h, 44

USE\_HAL\_QSPI\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 93, 106, 120  
  stm32f4xx\_hal\_conf\_C.h, 44

USE\_HAL\_RNG\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 93, 106, 120  
  stm32f4xx\_hal\_conf\_C.h, 44

USE\_HAL\_RTC\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 93, 106, 120  
  stm32f4xx\_hal\_conf\_C.h, 44

USE\_HAL\_SAI\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 93, 106, 120  
  stm32f4xx\_hal\_conf\_C.h, 44

USE\_HAL\_SD\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 93, 107, 120  
  stm32f4xx\_hal\_conf\_C.h, 44

USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 93, 107, 120  
  stm32f4xx\_hal\_conf\_C.h, 44

USE\_HAL\_SMARTCARD\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 93, 107, 121  
  stm32f4xx\_hal\_conf\_C.h, 45

USE\_HAL\_SMBUS\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 93, 107, 121  
  stm32f4xx\_hal\_conf\_C.h, 45

USE\_HAL\_SPDIFRX\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 94, 107, 121  
  stm32f4xx\_hal\_conf\_C.h, 45

USE\_HAL\_SPI\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 94, 107, 121  
  stm32f4xx\_hal\_conf\_C.h, 45

USE\_HAL\_SRAM\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 94, 107, 121  
  stm32f4xx\_hal\_conf\_C.h, 45

USE\_HAL\_TIM\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 94, 107, 121  
  stm32f4xx\_hal\_conf\_C.h, 45

USE\_HAL\_UART\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 94, 107, 121  
  stm32f4xx\_hal\_conf\_C.h, 45

USE\_HAL\_USART\_REGISTER\_CALLBACKS  
  stm32f4xx\_hal\_conf.h, 94, 107, 121  
  stm32f4xx\_hal\_conf\_C.h, 45

## USE\_HAL\_WWDG\_REGISTER\_CALLBACKS

stm32f4xx\_hal\_conf.h, [94](#), [108](#), [121](#)stm32f4xx\_hal\_conf\_C.h, [45](#)

## USE\_RTOS

stm32f4xx\_hal\_conf.h, [94](#), [108](#), [121](#)stm32f4xx\_hal\_conf\_C.h, [45](#)

## USE\_SPI\_CRC

stm32f4xx\_hal\_conf.h, [94](#), [108](#), [122](#)stm32f4xx\_hal\_conf\_C.h, [46](#)

## VDD\_VALUE

stm32f4xx\_hal\_conf.h, [94](#), [108](#), [122](#)stm32f4xx\_hal\_conf\_C.h, [46](#)