

# e-Tech Racing's Inverter Firmware

v0

Generated by Doxygen 1.10.0



---

<b>1 Topic Index</b>	<b>1</b>
1.1 Topics . . . . .	1
<b>2 Data Structure Index</b>	<b>3</b>
2.1 Data Structures . . . . .	3
<b>3 File Index</b>	<b>5</b>
3.1 File List . . . . .	5
<b>4 Topic Documentation</b>	<b>7</b>
4.1 Math Constants . . . . .	7
4.1.1 Detailed Description . . . . .	7
4.1.2 Macro Definition Documentation . . . . .	7
4.1.2.1 DIV2 . . . . .	7
4.1.2.2 INV3 . . . . .	7
4.1.2.3 INV_DEG . . . . .	8
4.1.2.4 IPI . . . . .	8
4.1.2.5 IPI2 . . . . .	8
4.1.2.6 ISQ2 . . . . .	8
4.1.2.7 ISQ3 . . . . .	8
4.1.2.8 PI . . . . .	8
4.1.2.9 PI2 . . . . .	8
4.1.2.10 SQ2 . . . . .	8
4.1.2.11 SQ3 . . . . .	9
4.2 - Integral Controllers . . . . .	9
4.2.1 Detailed Description . . . . .	9
4.2.2 Function Documentation . . . . .	9
4.2.2.1 pi_aw_calc() . . . . .	9
4.2.2.2 pi_calc() . . . . .	10
4.2.2.3 pi_extsat_calc() . . . . .	10
4.2.2.4 pi_init() . . . . .	11
4.3 Clarke and Park Transformations . . . . .	11
4.3.1 Detailed Description . . . . .	12
4.3.2 Function Documentation . . . . .	12
4.3.2.1 clarke3F_calc() . . . . .	12
4.3.2.2 iclarke3F_calc() . . . . .	13
4.3.2.3 irot_calc() . . . . .	13
4.3.2.4 rot_calc() . . . . .	14
4.4 Utility Functions . . . . .	14
4.4.1 Detailed Description . . . . .	15
4.4.2 Function Documentation . . . . .	15
4.4.2.1 angle_calc() . . . . .	15
4.4.2.2 svpwm_calc() . . . . .	15

---

4.5 Signal Processing Functions . . . . .	16
4.5.1 Detailed Description . . . . .	17
4.5.2 Macro Definition Documentation . . . . .	17
4.5.2.1 N_DATALOG . . . . .	17
4.5.3 Function Documentation . . . . .	17
4.5.3.1 avg_calc_10_samples() . . . . .	17
4.5.3.2 datalog_calc() . . . . .	17
4.5.3.3 filtreLP_calc() . . . . .	18
4.5.3.4 filtreLP_init() . . . . .	18
4.5.3.5 rampa_calc() . . . . .	19
4.5.3.6 rampa_dual_calc() . . . . .	19
4.5.3.7 RMS_calc() . . . . .	19
4.6 Miscellaneous Functions . . . . .	20
4.6.1 Detailed Description . . . . .	20
4.6.2 Function Documentation . . . . .	20
4.6.2.1 step_calc() . . . . .	20
4.7 CMSIS . . . . .	21
4.7.1 Detailed Description . . . . .	21
4.7.2 Stm32f7xx_system . . . . .	21
4.7.2.1 Detailed Description . . . . .	22
4.7.2.2 STM32F7xx_System_Private_Includes . . . . .	22
4.7.2.3 STM32F7xx_System_Private_TypesDefinitions . . . . .	23
4.7.2.4 STM32F7xx_System_Private_Defines . . . . .	23
4.7.2.5 STM32F7xx_System_Private_Macros . . . . .	23
4.7.2.6 STM32F7xx_System_Private_Variables . . . . .	24
4.7.2.7 STM32F7xx_System_Private_FunctionPrototypes . . . . .	24
4.7.2.8 STM32F7xx_System_Private_Functions . . . . .	25
<b>5 Data Structure Documentation</b> . . . . .	<b>27</b>
5.1 angle_struct Struct Reference . . . . .	27
5.1.1 Detailed Description . . . . .	27
5.1.2 Field Documentation . . . . .	27
5.1.2.1 angle . . . . .	27
5.1.2.2 calc . . . . .	27
5.1.2.3 freq . . . . .	28
5.1.2.4 Ts . . . . .	28
5.2 avg_struct_10 Struct Reference . . . . .	28
5.2.1 Detailed Description . . . . .	28
5.2.2 Field Documentation . . . . .	28
5.2.2.1 in . . . . .	28
5.2.2.2 out . . . . .	28
5.3 clarke3F_struct Struct Reference . . . . .	29

---

5.3.1 Detailed Description . . . . .	29
5.3.2 Field Documentation . . . . .	29
5.3.2.1 a . . . . .	29
5.3.2.2 b . . . . .	29
5.3.2.3 calc . . . . .	29
5.3.2.4 D . . . . .	29
5.3.2.5 Q . . . . .	30
5.4 Compare3F_struct Struct Reference . . . . .	30
5.4.1 Field Documentation . . . . .	30
5.4.1.1 alfaA . . . . .	30
5.4.1.2 alfaB . . . . .	30
5.4.1.3 alfaC . . . . .	30
5.4.1.4 calc . . . . .	30
5.4.1.5 compA . . . . .	31
5.4.1.6 compB . . . . .	31
5.4.1.7 compC . . . . .	31
5.5 datalog_struct Struct Reference . . . . .	31
5.5.1 Field Documentation . . . . .	31
5.5.1.1 calc . . . . .	31
5.5.1.2 estat . . . . .	31
5.5.1.3 i . . . . .	32
5.5.1.4 j . . . . .	32
5.5.1.5 log . . . . .	32
5.5.1.6 prescaler . . . . .	32
5.5.1.7 var . . . . .	32
5.6 Duties Struct Reference . . . . .	32
5.6.1 Detailed Description . . . . .	33
5.6.2 Field Documentation . . . . .	33
5.6.2.1 Da . . . . .	33
5.6.2.2 Db . . . . .	33
5.6.2.3 Dc . . . . .	33
5.7 Encoder Struct Reference . . . . .	33
5.7.1 Field Documentation . . . . .	34
5.7.1.1 A . . . . .	34
5.7.1.2 B . . . . .	34
5.7.1.3 DIR . . . . .	34
5.7.1.4 theta_e . . . . .	34
5.7.1.5 theta_m . . . . .	34
5.7.1.6 we . . . . .	34
5.7.1.7 wm_rpm . . . . .	34
5.7.1.8 Z . . . . .	35
5.8 filtreLP_struct Struct Reference . . . . .	35

5.8.1 Detailed Description . . . . .	35
5.8.2 Field Documentation . . . . .	35
5.8.2.1 alfa . . . . .	35
5.8.2.2 calc . . . . .	35
5.8.2.3 enable . . . . .	36
5.8.2.4 fc . . . . .	36
5.8.2.5 in . . . . .	36
5.8.2.6 init . . . . .	36
5.8.2.7 out . . . . .	36
5.8.2.8 Ts . . . . .	36
5.9 iclarke3F_struct Struct Reference . . . . .	36
5.9.1 Detailed Description . . . . .	37
5.9.2 Field Documentation . . . . .	37
5.9.2.1 a . . . . .	37
5.9.2.2 b . . . . .	37
5.9.2.3 calc . . . . .	37
5.9.2.4 D . . . . .	37
5.9.2.5 Q . . . . .	37
5.10 inverterStruct Struct Reference . . . . .	38
5.10.1 Detailed Description . . . . .	38
5.10.2 Field Documentation . . . . .	38
5.10.2.1 ADC_raw . . . . .	38
5.10.2.2 duties . . . . .	39
5.10.2.3 enable_pin . . . . .	39
5.10.2.4 enable_port . . . . .	39
5.10.2.5 encoder . . . . .	39
5.10.2.6 hadc . . . . .	39
5.10.2.7 htim . . . . .	39
5.10.2.8 led . . . . .	39
5.10.2.9 measurements . . . . .	39
5.10.2.10 state . . . . .	40
5.11 irot_struct Struct Reference . . . . .	40
5.11.1 Detailed Description . . . . .	40
5.11.2 Field Documentation . . . . .	40
5.11.2.1 calc . . . . .	40
5.11.2.2 cosFi . . . . .	40
5.11.2.3 d . . . . .	41
5.11.2.4 D . . . . .	41
5.11.2.5 q . . . . .	41
5.11.2.6 Q . . . . .	41
5.11.2.7 sinFi . . . . .	41
5.12 LED Struct Reference . . . . .	41

---

5.12.1 Detailed Description . . . . .	42
5.12.2 Field Documentation . . . . .	42
5.12.2.1 mode . . . . .	42
5.12.2.2 pin . . . . .	42
5.12.2.3 port . . . . .	42
5.13 Measurements Struct Reference . . . . .	42
5.13.1 Field Documentation . . . . .	42
5.13.1.1 ia . . . . .	42
5.13.1.2 ib . . . . .	43
5.13.1.3 ic . . . . .	43
5.13.1.4 VDC . . . . .	43
5.14 pi_aw_struct Struct Reference . . . . .	43
5.14.1 Detailed Description . . . . .	44
5.14.2 Field Documentation . . . . .	44
5.14.2.1 calc . . . . .	44
5.14.2.2 e . . . . .	44
5.14.2.3 enable . . . . .	44
5.14.2.4 Kaw . . . . .	44
5.14.2.5 Ki . . . . .	44
5.14.2.6 Kp . . . . .	44
5.14.2.7 pi_consig . . . . .	45
5.14.2.8 pi_fdb . . . . .	45
5.14.2.9 pi_ffw . . . . .	45
5.14.2.10 pi_int . . . . .	45
5.14.2.11 pi_out . . . . .	45
5.14.2.12 pi_out_max . . . . .	45
5.14.2.13 pi_out_min . . . . .	45
5.14.2.14 pi_out_postsat . . . . .	45
5.14.2.15 pi_out_presat . . . . .	46
5.14.2.16 Ts . . . . .	46
5.15 pi_struct Struct Reference . . . . .	46
5.15.1 Detailed Description . . . . .	46
5.15.2 Field Documentation . . . . .	47
5.15.2.1 calc . . . . .	47
5.15.2.2 e . . . . .	47
5.15.2.3 enable . . . . .	47
5.15.2.4 init . . . . .	47
5.15.2.5 K0 . . . . .	47
5.15.2.6 K1 . . . . .	47
5.15.2.7 Ki . . . . .	47
5.15.2.8 Kp . . . . .	48
5.15.2.9 pi_consig . . . . .	48

5.15.2.10 pi_fdb . . . . .	48
5.15.2.11 pi_ffw . . . . .	48
5.15.2.12 pi_out . . . . .	48
5.15.2.13 pi_out_max . . . . .	48
5.15.2.14 pi_out_min . . . . .	48
5.15.2.15 Ts . . . . .	49
5.16 rampa_dual_struct Struct Reference . . . . .	49
5.16.1 Detailed Description . . . . .	49
5.16.2 Field Documentation . . . . .	49
5.16.2.1 calc . . . . .	49
5.16.2.2 Decr . . . . .	49
5.16.2.3 enable . . . . .	50
5.16.2.4 in . . . . .	50
5.16.2.5 Incr . . . . .	50
5.16.2.6 out . . . . .	50
5.17 rampa_struct Struct Reference . . . . .	50
5.17.1 Detailed Description . . . . .	50
5.17.2 Field Documentation . . . . .	51
5.17.2.1 calc . . . . .	51
5.17.2.2 enable . . . . .	51
5.17.2.3 in . . . . .	51
5.17.2.4 Incr . . . . .	51
5.17.2.5 out . . . . .	51
5.18 RMS_struct Struct Reference . . . . .	51
5.18.1 Detailed Description . . . . .	52
5.18.2 Field Documentation . . . . .	52
5.18.2.1 Angle . . . . .	52
5.18.2.2 Angle_ant . . . . .	52
5.18.2.3 Freq . . . . .	52
5.18.2.4 Measure . . . . .	52
5.18.2.5 Out_RMS . . . . .	52
5.18.2.6 Sq_Sum . . . . .	53
5.18.2.7 T_exec . . . . .	53
5.19 rot_struct Struct Reference . . . . .	53
5.19.1 Detailed Description . . . . .	53
5.19.2 Field Documentation . . . . .	53
5.19.2.1 calc . . . . .	53
5.19.2.2 cosFi . . . . .	54
5.19.2.3 D . . . . .	54
5.19.2.4 d . . . . .	54
5.19.2.5 Q . . . . .	54
5.19.2.6 q . . . . .	54

---

5.19.2.7 sinFi . . . . .	54
5.20 step_struct Struct Reference . . . . .	54
5.20.1 Detailed Description . . . . .	55
5.20.2 Field Documentation . . . . .	55
5.20.2.1 calc . . . . .	55
5.20.2.2 Counter . . . . .	55
5.20.2.3 enable . . . . .	55
5.20.2.4 fs . . . . .	55
5.20.2.5 In . . . . .	55
5.20.2.6 Out . . . . .	56
5.20.2.7 Pulses . . . . .	56
5.20.2.8 Step . . . . .	56
5.20.2.9 t_step . . . . .	56
5.21 svpwm_struct Struct Reference . . . . .	56
5.21.1 Detailed Description . . . . .	56
5.21.2 Field Documentation . . . . .	57
5.21.2.1 calc . . . . .	57
5.21.2.2 Ta . . . . .	57
5.21.2.3 Tb . . . . .	57
5.21.2.4 Tc . . . . .	57
5.21.2.5 valfa . . . . .	57
5.21.2.6 vbeta . . . . .	57
<b>6 File Documentation</b> . . . . .	<b>59</b>
6.1 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/adc.h File Reference . . . . .	59
6.1.1 Detailed Description . . . . .	60
6.1.2 Function Documentation . . . . .	61
6.1.2.1 MX_ADC1_Init() . . . . .	61
6.1.2.2 MX_ADC2_Init() . . . . .	61
6.1.2.3 MX_ADC3_Init() . . . . .	62
6.1.3 Variable Documentation . . . . .	63
6.1.3.1 hadc1 . . . . .	63
6.1.3.2 hadc2 . . . . .	63
6.1.3.3 hadc3 . . . . .	63
6.2 adc.h . . . . .	63
6.3 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/can.h File Reference . . . . .	64
6.3.1 Detailed Description . . . . .	65
6.3.2 Function Documentation . . . . .	65
6.3.2.1 MX_CAN1_Init() . . . . .	65
6.3.3 Variable Documentation . . . . .	66
6.3.3.1 hcan1 . . . . .	66
6.4 can.h . . . . .	66

6.5 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/CONTROL.h File Reference . . . . .	66
6.5.1 Detailed Description . . . . .	67
6.5.2 Function Documentation . . . . .	68
6.5.2.1 control() . . . . .	68
6.5.3 Variable Documentation . . . . .	69
6.5.3.1 valfa_L . . . . .	69
6.5.3.2 vbeta_L . . . . .	69
6.6 CONTROL.h . . . . .	69
6.7 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/dac.h File Reference . . . . .	69
6.7.1 Detailed Description . . . . .	70
6.7.2 Function Documentation . . . . .	71
6.7.2.1 MX_DAC_Init() . . . . .	71
6.7.3 Variable Documentation . . . . .	71
6.7.3.1 hdac . . . . .	71
6.8 dac.h . . . . .	71
6.9 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/dma.h File Reference . . . . .	72
6.9.1 Detailed Description . . . . .	73
6.9.2 Function Documentation . . . . .	73
6.9.2.1 MX_DMA_Init() . . . . .	73
6.10 dma.h . . . . .	74
6.11 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/FSM.h File Reference . . . . .	74
6.11.1 Detailed Description . . . . .	76
6.11.2 Function Documentation . . . . .	76
6.11.2.1 inv_FSM() . . . . .	76
6.12 FSM.h . . . . .	76
6.13 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/gpio.h File Reference . . . . .	77
6.13.1 Detailed Description . . . . .	78
6.13.2 Function Documentation . . . . .	78
6.13.2.1 MX_GPIO_Init() . . . . .	78
6.14 gpio.h . . . . .	79
6.15 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/i2c.h File Reference . . . . .	79
6.15.1 Detailed Description . . . . .	80
6.15.2 Function Documentation . . . . .	81
6.15.2.1 MX_I2C1_Init() . . . . .	81
6.15.3 Variable Documentation . . . . .	81
6.15.3.1 hi2c1 . . . . .	81
6.16 i2c.h . . . . .	81
6.17 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/INVERTER.h File Reference . . . . .	82
6.17.1 Detailed Description . . . . .	83
6.17.2 Enumeration Type Documentation . . . . .	84
6.17.2.1 inverterState . . . . .	84
6.17.3 Function Documentation . . . . .	84

---

6.17.3.1 inv_init()	84
6.17.4 Variable Documentation	85
6.17.4.1 invLeft	85
6.17.4.2 invRight	85
6.18 INVERTER.h	85
6.19 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/main.h File Reference	86
6.19.1 Detailed Description	88
6.19.2 Macro Definition Documentation	89
6.19.2.1 A_L_GPIO_Port	89
6.19.2.2 A_L_Pin	89
6.19.2.3 A_R_GPIO_Port	89
6.19.2.4 A_R_Pin	89
6.19.2.5 B_L_GPIO_Port	89
6.19.2.6 B_L_Pin	89
6.19.2.7 B_R_GPIO_Port	89
6.19.2.8 B_R_Pin	89
6.19.2.9 DAC_GPIO_Port	89
6.19.2.10 DAC_Pin	89
6.19.2.11 DIR_GPIO_Port	90
6.19.2.12 DIR_Pin	90
6.19.2.13 DT	90
6.19.2.14 ENABLE_L_GPIO_Port	90
6.19.2.15 ENABLE_L_Pin	90
6.19.2.16 ENABLE_R_GPIO_Port	90
6.19.2.17 ENABLE_R_Pin	90
6.19.2.18 ia_L_GPIO_Port	90
6.19.2.19 ia_L_Pin	90
6.19.2.20 ia_R_GPIO_Port	90
6.19.2.21 ia_R_Pin	91
6.19.2.22 ib_L_GPIO_Port	91
6.19.2.23 ib_L_Pin	91
6.19.2.24 ib_R_GPIO_Port	91
6.19.2.25 ib_R_Pin	91
6.19.2.26 ic_L_GPIO_Port	91
6.19.2.27 ic_L_Pin	91
6.19.2.28 ic_R_GPIO_Port	91
6.19.2.29 ic_R_Pin	91
6.19.2.30 LED_ERR_GPIO_Port	91
6.19.2.31 LED_ERR_Pin	92
6.19.2.32 LED_LEFT_GPIO_Port	92
6.19.2.33 LED_LEFT_Pin	92
6.19.2.34 LED_RIGHT_GPIO_Port	92

---

6.19.2.35 LED_RIGHT_Pin . . . . .	92
6.19.2.36 PWM1_L_GPIO_Port . . . . .	92
6.19.2.37 PWM1_L_Pin . . . . .	92
6.19.2.38 PWM1_R_GPIO_Port . . . . .	92
6.19.2.39 PWM1_R_Pin . . . . .	92
6.19.2.40 PWM2_L_GPIO_Port . . . . .	92
6.19.2.41 PWM2_L_Pin . . . . .	93
6.19.2.42 PWM2_R_GPIO_Port . . . . .	93
6.19.2.43 PWM2_R_Pin . . . . .	93
6.19.2.44 PWM3_L_GPIO_Port . . . . .	93
6.19.2.45 PWM3_L_Pin . . . . .	93
6.19.2.46 PWM3_R_GPIO_Port . . . . .	93
6.19.2.47 PWM3_R_Pin . . . . .	93
6.19.2.48 PWM4_L_GPIO_Port . . . . .	93
6.19.2.49 PWM4_L_Pin . . . . .	93
6.19.2.50 PWM4_R_GPIO_Port . . . . .	93
6.19.2.51 PWM4_R_Pin . . . . .	94
6.19.2.52 PWM5_L_GPIO_Port . . . . .	94
6.19.2.53 PWM5_L_Pin . . . . .	94
6.19.2.54 PWM5_R_GPIO_Port . . . . .	94
6.19.2.55 PWM5_R_Pin . . . . .	94
6.19.2.56 PWM6_L_GPIO_Port . . . . .	94
6.19.2.57 PWM6_L_Pin . . . . .	94
6.19.2.58 PWM6_R_GPIO_Port . . . . .	94
6.19.2.59 PWM6_R_Pin . . . . .	94
6.19.2.60 SC_det_GPIO_Port . . . . .	94
6.19.2.61 SC_det_Pin . . . . .	95
6.19.2.62 Tinv_L_GPIO_Port . . . . .	95
6.19.2.63 Tinv_L_Pin . . . . .	95
6.19.2.64 Tinv_R_GPIO_Port . . . . .	95
6.19.2.65 Tinv_R_Pin . . . . .	95
6.19.2.66 Tmot_L_GPIO_Port . . . . .	95
6.19.2.67 Tmot_L_Pin . . . . .	95
6.19.2.68 Tmot_R_GPIO_Port . . . . .	95
6.19.2.69 Tmot_R_Pin . . . . .	95
6.19.2.70 TRIP_L_GPIO_Port . . . . .	95
6.19.2.71 TRIP_L_Pin . . . . .	96
6.19.2.72 TRIP_R_GPIO_Port . . . . .	96
6.19.2.73 TRIP_R_Pin . . . . .	96
6.19.2.74 TS . . . . .	96
6.19.2.75 VDC_L_GPIO_Port . . . . .	96
6.19.2.76 VDC_L_Pin . . . . .	96

---

6.19.2.77 VDC_R_GPIO_Port . . . . .	96
6.19.2.78 VDC_R_Pin . . . . .	96
6.19.2.79 WRN_L_GPIO_Port . . . . .	96
6.19.2.80 WRN_L_Pin . . . . .	96
6.19.2.81 WRN_R_GPIO_Port . . . . .	97
6.19.2.82 WRN_R_Pin . . . . .	97
6.19.2.83 Z_L_GPIO_Port . . . . .	97
6.19.2.84 Z_L_Pin . . . . .	97
6.19.2.85 Z_R_GPIO_Port . . . . .	97
6.19.2.86 Z_R_Pin . . . . .	97
6.19.3 Function Documentation . . . . .	97
6.19.3.1 Error_Handler() . . . . .	97
6.20 main.h . . . . .	98
6.21 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/MEASUREMENTS.h File Reference . . . . .	100
6.21.1 Detailed Description . . . . .	101
6.21.2 Macro Definition Documentation . . . . .	102
6.21.2.1 CURRENT_OFFSET . . . . .	102
6.21.2.2 CURRENT_SLOPE . . . . .	102
6.21.2.3 VOLTAGE_OFFSET . . . . .	102
6.21.2.4 VOLTAGE_SLOPE . . . . .	102
6.21.3 Function Documentation . . . . .	102
6.21.3.1 getADCelec() . . . . .	102
6.21.3.2 getLinear() . . . . .	103
6.22 MEASUREMENTS.h . . . . .	104
6.23 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PCB_IO.h File Reference . . . . .	104
6.23.1 Detailed Description . . . . .	106
6.23.2 Macro Definition Documentation . . . . .	106
6.23.2.1 DIR_STATE . . . . .	106
6.23.2.2 DISABLE . . . . .	106
6.23.2.3 ENABLE . . . . .	106
6.23.2.4 SC_DET_STATE . . . . .	107
6.23.2.5 WRN_STATE . . . . .	107
6.23.3 Enumeration Type Documentation . . . . .	107
6.23.3.1 LEDMode . . . . .	107
6.23.4 Function Documentation . . . . .	107
6.23.4.1 LED_handler() . . . . .	107
6.23.5 Variable Documentation . . . . .	108
6.23.5.1 led_error . . . . .	108
6.23.5.2 led_left . . . . .	108
6.23.5.3 led_right . . . . .	108
6.24 PCB_IO.h . . . . .	108
6.25 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h File Reference . . . . .	109

---

6.26 Pergamon_float.h . . . . .	111
6.27 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PWM.h File Reference . . . . .	114
6.27.1 Detailed Description . . . . .	116
6.27.2 Function Documentation . . . . .	116
6.27.2.1 Compare3F_calc() . . . . .	116
6.27.2.2 disable_PWM() . . . . .	116
6.27.2.3 enable_PWM() . . . . .	117
6.27.2.4 set_PWM() . . . . .	117
6.27.3 Variable Documentation . . . . .	118
6.27.3.1 compares_LEFT . . . . .	118
6.27.3.2 compares_RIGHT . . . . .	118
6.28 PWM.h . . . . .	118
6.29 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/stm32f7xx_hal_conf.h File Reference . . . . .	119
6.29.1 Macro Definition Documentation . . . . .	121
6.29.1.1 ART_ACCELERATOR_ENABLE . . . . .	121
6.29.1.2 assert_param . . . . .	122
6.29.1.3 DP83848_PHY_ADDRESS . . . . .	122
6.29.1.4 ETH_RX_BUF_SIZE . . . . .	122
6.29.1.5 ETH_RXBUFN B . . . . .	122
6.29.1.6 ETH_TX_BUF_SIZE . . . . .	122
6.29.1.7 ETH_TXBUFN B . . . . .	122
6.29.1.8 EXTERNAL_CLOCK_VALUE . . . . .	122
6.29.1.9 HAL_ADC_MODULE_ENABLED . . . . .	122
6.29.1.10 HAL_CAN_MODULE_ENABLED . . . . .	122
6.29.1.11 HAL_CORTEX_MODULE_ENABLED . . . . .	123
6.29.1.12 HAL_DAC_MODULE_ENABLED . . . . .	123
6.29.1.13 HAL_DMA_MODULE_ENABLED . . . . .	123
6.29.1.14 HAL_EXTI_MODULE_ENABLED . . . . .	123
6.29.1.15 HAL_FLASH_MODULE_ENABLED . . . . .	123
6.29.1.16 HAL_GPIO_MODULE_ENABLED . . . . .	123
6.29.1.17 HAL_I2C_MODULE_ENABLED . . . . .	123
6.29.1.18 HAL_MODULE_ENABLED . . . . .	123
6.29.1.19 HAL_PWR_MODULE_ENABLED . . . . .	123
6.29.1.20 HAL_RCC_MODULE_ENABLED . . . . .	123
6.29.1.21 HAL_TIM_MODULE_ENABLED . . . . .	124
6.29.1.22 HSE_STARTUP_TIMEOUT . . . . .	124
6.29.1.23 HSE_VALUE . . . . .	124
6.29.1.24 HSI_VALUE . . . . .	124
6.29.1.25 LSE_STARTUP_TIMEOUT . . . . .	124
6.29.1.26 LSE_VALUE . . . . .	124
6.29.1.27 LSI_VALUE . . . . .	125
6.29.1.28 MAC_ADDR0 . . . . .	125

---

6.29.1.29 MAC_ADDR1 . . . . .	125
6.29.1.30 MAC_ADDR2 . . . . .	125
6.29.1.31 MAC_ADDR3 . . . . .	125
6.29.1.32 MAC_ADDR4 . . . . .	125
6.29.1.33 MAC_ADDR5 . . . . .	125
6.29.1.34 PHY_AUTONEGO_COMPLETE . . . . .	125
6.29.1.35 PHY_AUTONEGOTIATION . . . . .	126
6.29.1.36 PHY_BCR . . . . .	126
6.29.1.37 PHY_BSR . . . . .	126
6.29.1.38 PHY_CONFIG_DELAY . . . . .	126
6.29.1.39 PHY_DUPLEX_STATUS . . . . .	126
6.29.1.40 PHY_FULLDUPLEX_100M . . . . .	126
6.29.1.41 PHY_FULLDUPLEX_10M . . . . .	126
6.29.1.42 PHY_HALFDUPLEX_100M . . . . .	127
6.29.1.43 PHY_HALFDUPLEX_10M . . . . .	127
6.29.1.44 PHY_ISOLATE . . . . .	127
6.29.1.45 PHY_JABBER_DETECTION . . . . .	127
6.29.1.46 PHY_LINKED_STATUS . . . . .	127
6.29.1.47 PHY_LOOPBACK . . . . .	127
6.29.1.48 PHY_POWERDOWN . . . . .	127
6.29.1.49 PHY_READ_TO . . . . .	128
6.29.1.50 PHY_RESET . . . . .	128
6.29.1.51 PHY_RESET_DELAY . . . . .	128
6.29.1.52 PHY_RESTART_AUTONEGOTIATION . . . . .	128
6.29.1.53 PHY_SPEED_STATUS . . . . .	128
6.29.1.54 PHY_SR . . . . .	128
6.29.1.55 PHY_WRITE_TO . . . . .	128
6.29.1.56 PREFETCH_ENABLE . . . . .	128
6.29.1.57 TICK_INT_PRIORITY . . . . .	129
6.29.1.58 USE_HAL_ADC_REGISTER_CALLBACKS . . . . .	129
6.29.1.59 USE_HAL_CAN_REGISTER_CALLBACKS . . . . .	129
6.29.1.60 USE_HAL_CEC_REGISTER_CALLBACKS . . . . .	129
6.29.1.61 USE_HAL_CRYP_REGISTER_CALLBACKS . . . . .	129
6.29.1.62 USE_HAL_DAC_REGISTER_CALLBACKS . . . . .	129
6.29.1.63 USE_HAL_DCMI_REGISTER_CALLBACKS . . . . .	129
6.29.1.64 USE_HAL_DFSDM_REGISTER_CALLBACKS . . . . .	129
6.29.1.65 USE_HAL_DMA2D_REGISTER_CALLBACKS . . . . .	129
6.29.1.66 USE_HAL_DSI_REGISTER_CALLBACKS . . . . .	129
6.29.1.67 USE_HAL_ETH_REGISTER_CALLBACKS . . . . .	130
6.29.1.68 USE_HAL_HASH_REGISTER_CALLBACKS . . . . .	130
6.29.1.69 USE_HAL_HCD_REGISTER_CALLBACKS . . . . .	130
6.29.1.70 USE_HAL_I2C_REGISTER_CALLBACKS . . . . .	130

---

6.29.1.71 USE_HAL_I2S_REGISTER_CALLBACKS . . . . .	130
6.29.1.72 USE_HAL_IRDA_REGISTER_CALLBACKS . . . . .	130
6.29.1.73 USE_HAL_JPEG_REGISTER_CALLBACKS . . . . .	130
6.29.1.74 USE_HAL_LPTIM_REGISTER_CALLBACKS . . . . .	130
6.29.1.75 USE_HAL_LTDC_REGISTER_CALLBACKS . . . . .	130
6.29.1.76 USE_HAL_MDIOS_REGISTER_CALLBACKS . . . . .	130
6.29.1.77 USE_HAL_MMC_REGISTER_CALLBACKS . . . . .	131
6.29.1.78 USE_HAL_NAND_REGISTER_CALLBACKS . . . . .	131
6.29.1.79 USE_HAL_NOR_REGISTER_CALLBACKS . . . . .	131
6.29.1.80 USE_HAL_PCD_REGISTER_CALLBACKS . . . . .	131
6.29.1.81 USE_HAL_QSPI_REGISTER_CALLBACKS . . . . .	131
6.29.1.82 USE_HAL_RNG_REGISTER_CALLBACKS . . . . .	131
6.29.1.83 USE_HAL_RTC_REGISTER_CALLBACKS . . . . .	131
6.29.1.84 USE_HAL_SAI_REGISTER_CALLBACKS . . . . .	131
6.29.1.85 USE_HAL_SD_REGISTER_CALLBACKS . . . . .	131
6.29.1.86 USE_HAL_SDRAM_REGISTER_CALLBACKS . . . . .	131
6.29.1.87 USE_HAL_SMARTCARD_REGISTER_CALLBACKS . . . . .	132
6.29.1.88 USE_HAL_SMBUS_REGISTER_CALLBACKS . . . . .	132
6.29.1.89 USE_HAL_SPDIFRX_REGISTER_CALLBACKS . . . . .	132
6.29.1.90 USE_HAL_SPI_REGISTER_CALLBACKS . . . . .	132
6.29.1.91 USE_HAL_SRAM_REGISTER_CALLBACKS . . . . .	132
6.29.1.92 USE_HAL_TIM_REGISTER_CALLBACKS . . . . .	132
6.29.1.93 USE_HAL_UART_REGISTER_CALLBACKS . . . . .	132
6.29.1.94 USE_HAL_USART_REGISTER_CALLBACKS . . . . .	132
6.29.1.95 USE_HAL_WWDG_REGISTER_CALLBACKS . . . . .	132
6.29.1.96 USE_RTOS . . . . .	132
6.29.1.97 USE_SPI_CRC . . . . .	133
6.29.1.98 VDD_VALUE . . . . .	133
6.30 stm32f7xx_hal_conf.h . . . . .	133
6.31 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/stm32f7xx_it.h File Reference . . . . .	138
6.31.1 Detailed Description . . . . .	139
6.31.2 Function Documentation . . . . .	139
6.31.2.1 BusFault_Handler() . . . . .	139
6.31.2.2 CAN1_RX0_IRQHandler() . . . . .	139
6.31.2.3 CAN1_RX1_IRQHandler() . . . . .	140
6.31.2.4 DebugMon_Handler() . . . . .	140
6.31.2.5 DMA2_Stream1_IRQHandler() . . . . .	140
6.31.2.6 DMA2_Stream2_IRQHandler() . . . . .	140
6.31.2.7 HardFault_Handler() . . . . .	140
6.31.2.8 MemManage_Handler() . . . . .	140
6.31.2.9 NMI_Handler() . . . . .	140
6.31.2.10 PendSV_Handler() . . . . .	141

---

6.31.2.11 SVC_Handler()	141
6.31.2.12 SysTick_Handler()	141
6.31.2.13 TIM1_UP_TIM10_IRQHandler()	141
6.31.2.14 TIM6_DAC_IRQHandler()	142
6.31.2.15 UsageFault_Handler()	142
6.32 stm32f7xx_it.h	142
6.33 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/TASKS_1ms.h File Reference	143
6.33.1 Detailed Description	143
6.33.2 Function Documentation	144
6.33.2.1 tasks_1ms()	144
6.34 TASKS_1ms.h	144
6.35 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/tim.h File Reference	145
6.35.1 Detailed Description	146
6.35.2 Function Documentation	146
6.35.2.1 HAL_TIM_MspPostInit()	146
6.35.2.2 MX_TIM1_Init()	147
6.35.2.3 MX_TIM2_Init()	147
6.35.2.4 MX_TIM4_Init()	148
6.35.2.5 MX_TIM6_Init()	148
6.35.2.6 MX_TIM8_Init()	149
6.35.3 Variable Documentation	150
6.35.3.1 htim1	150
6.35.3.2 htim2	150
6.35.3.3 htim4	150
6.35.3.4 htim6	150
6.35.3.5 htim8	150
6.36 tim.h	150
6.37 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/usb_otg.h File Reference	151
6.37.1 Detailed Description	152
6.37.2 Function Documentation	152
6.37.2.1 MX_USB_OTG_FS_USB_Init()	152
6.38 usb_otg.h	153
6.39 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/adc.c File Reference	153
6.39.1 Detailed Description	154
6.39.2 Function Documentation	155
6.39.2.1 HAL_ADC_MspDeInit()	155
6.39.2.2 HAL_ADC_MspInit()	155
6.39.2.3 MX_ADC1_Init()	155
6.39.2.4 MX_ADC2_Init()	156
6.39.2.5 MX_ADC3_Init()	157
6.39.3 Variable Documentation	157
6.39.3.1 hadc1	157

---

6.39.3.2 hadc2 . . . . .	158
6.39.3.3 hadc3 . . . . .	158
6.39.3.4 hdma_adc2 . . . . .	158
6.39.3.5 hdma_adc3 . . . . .	158
6.40 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/can.c File Reference . . . . .	158
6.40.1 Detailed Description . . . . .	159
6.40.2 Function Documentation . . . . .	159
6.40.2.1 HAL_CAN_MspDelInit() . . . . .	159
6.40.2.2 HAL_CAN_MspInit() . . . . .	159
6.40.2.3 MX_CAN1_Init() . . . . .	160
6.40.3 Variable Documentation . . . . .	160
6.40.3.1 hcan1 . . . . .	160
6.41 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/CONTROL.c File Reference . . . . .	160
6.41.1 Detailed Description . . . . .	161
6.41.2 Function Documentation . . . . .	161
6.41.2.1 control() . . . . .	161
6.41.3 Variable Documentation . . . . .	162
6.41.3.1 valfa_L . . . . .	162
6.41.3.2 vbeta_L . . . . .	162
6.42 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/dac.c File Reference . . . . .	162
6.42.1 Detailed Description . . . . .	163
6.42.2 Function Documentation . . . . .	164
6.42.2.1 HAL_DAC_MspDelInit() . . . . .	164
6.42.2.2 HAL_DAC_MspInit() . . . . .	164
6.42.2.3 MX_DAC_Init() . . . . .	164
6.42.3 Variable Documentation . . . . .	165
6.42.3.1 hdac . . . . .	165
6.43 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/dma.c File Reference . . . . .	165
6.43.1 Detailed Description . . . . .	165
6.43.2 Function Documentation . . . . .	166
6.43.2.1 MX_DMA_Init() . . . . .	166
6.44 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/FSM.c File Reference . . . . .	166
6.44.1 Detailed Description . . . . .	167
6.44.2 Function Documentation . . . . .	167
6.44.2.1 inv_FSM() . . . . .	167
6.45 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/gpio.c File Reference . . . . .	168
6.45.1 Detailed Description . . . . .	168
6.45.2 Function Documentation . . . . .	169
6.45.2.1 MX_GPIO_Init() . . . . .	169
6.46 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/i2c.c File Reference . . . . .	169
6.46.1 Detailed Description . . . . .	170
6.46.2 Function Documentation . . . . .	170

---

6.46.2.1 HAL_I2C_MspDeInit()	170
6.46.2.2 HAL_I2C_MspInit()	170
6.46.2.3 MX_I2C1_Init()	171
6.46.3 Variable Documentation	171
6.46.3.1 hi2c1	171
6.47 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/INVERTER.c File Reference	171
6.47.1 Detailed Description	172
6.47.2 Function Documentation	173
6.47.2.1 inv_init()	173
6.47.3 Variable Documentation	173
6.47.3.1 invLeft	173
6.47.3.2 invRight	174
6.48 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/main.c File Reference	174
6.48.1 Detailed Description	175
6.48.2 Function Documentation	175
6.48.2.1 Error_Handler()	175
6.48.2.2 main()	176
6.48.2.3 SystemClock_Config()	177
6.49 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/MEASUREMENTS.c File Reference	178
6.49.1 Detailed Description	179
6.49.2 Function Documentation	179
6.49.2.1 getADCelec()	179
6.49.2.2 getLinear()	180
6.50 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/PCB_IO.c File Reference	181
6.50.1 Detailed Description	182
6.50.2 Function Documentation	182
6.50.2.1 LED_handler()	182
6.50.3 Variable Documentation	183
6.50.3.1 led_error	183
6.50.3.2 led_left	183
6.50.3.3 led_right	183
6.51 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/Pergamon_float.c File Reference	183
6.52 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/PWM.c File Reference	184
6.52.1 Detailed Description	185
6.52.2 Function Documentation	186
6.52.2.1 Compare3F_calc()	186
6.52.2.2 disable_PWM()	186
6.52.2.3 enable_PWM()	186
6.52.2.4 set_PWM()	187
6.53 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/stm32f7xx_hal_msp.c File Reference	188
6.53.1 Detailed Description	188
6.53.2 Function Documentation	188

---

6.53.2.1 HAL_MspInit()	188
6.54 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/stm32f7xx_it.c File Reference	189
6.54.1 Detailed Description	190
6.54.2 Function Documentation	190
6.54.2.1 BusFault_Handler()	190
6.54.2.2 CAN1_RX0_IRQHandler()	190
6.54.2.3 CAN1_RX1_IRQHandler()	191
6.54.2.4 DebugMon_Handler()	191
6.54.2.5 DMA2_Stream1_IRQHandler()	191
6.54.2.6 DMA2_Stream2_IRQHandler()	191
6.54.2.7 HardFault_Handler()	191
6.54.2.8 MemManage_Handler()	191
6.54.2.9 NMI_Handler()	191
6.54.2.10 PendSV_Handler()	192
6.54.2.11 SVC_Handler()	192
6.54.2.12 SysTick_Handler()	192
6.54.2.13 TIM1_UP_TIM10_IRQHandler()	192
6.54.2.14 TIM6_DAC_IRQHandler()	193
6.54.2.15 UsageFault_Handler()	193
6.54.3 Variable Documentation	193
6.54.3.1 hcan1	193
6.54.3.2 hdac	193
6.54.3.3 hdma_adc2	193
6.54.3.4 hdma_adc3	193
6.54.3.5 htim1	193
6.54.3.6 htim6	194
6.55 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/syscalls.c File Reference	194
6.55.1 Detailed Description	195
6.55.2 Function Documentation	195
6.55.2.1 __attribute__()	195
6.55.2.2 __io_getchar()	196
6.55.2.3 __io_putchar()	196
6.55.2.4 _close()	196
6.55.2.5 _execve()	196
6.55.2.6 _exit()	196
6.55.2.7 _fork()	197
6.55.2.8 _fstat()	197
6.55.2.9 _getpid()	197
6.55.2.10 _isatty()	197
6.55.2.11 _kill()	197
6.55.2.12 _link()	197
6.55.2.13 _lseek()	198

---

6.55.2.14 _open()	198
6.55.2.15 _stat()	198
6.55.2.16 _times()	198
6.55.2.17 _unlink()	198
6.55.2.18 _wait()	198
6.55.2.19 initialise_monitor_handles()	198
6.55.3 Variable Documentation	198
6.55.3.1 environ	198
6.56 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/sysmem.c File Reference	199
6.56.1 Detailed Description	199
6.56.2 Function Documentation	200
6.56.2.1 _sbrk()	200
6.57 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/system_stm32f7xx.c File Reference	200
6.57.1 Detailed Description	201
6.58 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/TASKS_1ms.c File Reference	202
6.58.1 Detailed Description	202
6.58.2 Function Documentation	203
6.58.2.1 tasks_1ms()	203
6.59 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/tim.c File Reference	203
6.59.1 Detailed Description	205
6.59.2 Function Documentation	205
6.59.2.1 HAL_TIM_Base_MspDeInit()	205
6.59.2.2 HAL_TIM_Base_MspInit()	205
6.59.2.3 HAL_TIM_IC_MspDeInit()	205
6.59.2.4 HAL_TIM_IC_MspInit()	205
6.59.2.5 HAL_TIM_MspPostInit()	206
6.59.2.6 MX_TIM1_Init()	206
6.59.2.7 MX_TIM2_Init()	207
6.59.2.8 MX_TIM4_Init()	207
6.59.2.9 MX_TIM6_Init()	208
6.59.2.10 MX_TIM8_Init()	208
6.59.3 Variable Documentation	209
6.59.3.1 htim1	209
6.59.3.2 htim2	209
6.59.3.3 htim4	209
6.59.3.4 htim6	209
6.59.3.5 htim8	209
6.60 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/usb_otg.c File Reference	210
6.60.1 Detailed Description	210
6.60.2 Function Documentation	211
6.60.2.1 MX_USB_OTG_FS_USB_Init()	211



# Chapter 1

## Topic Index

### 1.1 Topics

Here is a list of all topics with brief descriptions:

Math Constants . . . . .	7
- Integral Controllers . . . . .	9
Clarke and Park Transformations . . . . .	11
Utility Functions . . . . .	14
Signal Processing Functions . . . . .	16
Miscellaneous Functions . . . . .	20
CMSIS . . . . .	21
Stm32f7xx_system . . . . .	21
STM32F7xx_System_Private_Includes . . . . .	22
STM32F7xx_System_Private_TypesDefinitions . . . . .	23
STM32F7xx_System_Private_Defines . . . . .	23
STM32F7xx_System_Private_Macros . . . . .	23
STM32F7xx_System_Private_Variables . . . . .	24
STM32F7xx_System_Private_FunctionPrototypes . . . . .	24
STM32F7xx_System_Private_Functions . . . . .	25



# Chapter 2

## Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

<code>angle_struct</code>	Generates an angle based on a fixed frequency . . . . .	27
<code>avg_struct_10</code>	Moving average filter for 10 samples . . . . .	28
<code>clarke3F_struct</code>	Clarke transformation for three-phase systems . . . . .	29
<code>Compare3F_struct</code>	. . . . .	30
<code>datalog_struct</code>	. . . . .	31
<code>Duties</code>	Structure to hold PWM configuration parameters . . . . .	32
<code>Encoder</code>	. . . . .	33
<code>filtreLP_struct</code>	First-order low-pass filter . . . . .	35
<code>iclarke3F_struct</code>	Inverse Clarke transformation for three-phase systems . . . . .	36
<code>inverterStruct</code>	Inverter structure . . . . .	38
<code>irot_struct</code>	Inverse rotation (counterclockwise) . . . . .	40
<code>LED</code>	LED structure . . . . .	41
<code>Measurements</code>	. . . . .	42
<code>pi_aw_struct</code>	PI Controller with internal saturation, anti-windup, and feedforward . . . . .	43
<code>pi_struct</code>	PI Controller with external saturation and feedforward . . . . .	46
<code>rampa_dual_struct</code>	Dual-ramp generator . . . . .	49
<code>rampa_struct</code>	Single-ramp generator . . . . .	50
<code>RMS_struct</code>	Root Mean Square (RMS) calculation . . . . .	51
<code>rot_struct</code>	Rotates the DQ axis in the opposite direction (clockwise) . . . . .	53
<code>step_struct</code>	Step function generator . . . . .	54
<code>svpwm_struct</code>	Space Vector Pulse Width Modulation (SVPWM) implementation . . . . .	56



# Chapter 3

## File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">adc.h</a>	59
This file contains all the function prototypes for the <a href="#">adc.c</a> file	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">can.h</a>	64
This file contains all the function prototypes for the <a href="#">can.c</a> file	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">CONTROL.h</a>	66
Header file for control logic	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">dac.h</a>	69
This file contains all the function prototypes for the <a href="#">dac.c</a> file	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">dma.h</a>	72
This file contains all the function prototypes for the <a href="#">dma.c</a> file	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">FSM.h</a>	74
Header file for Finite State Machine (FSM) control	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">gpio.h</a>	77
This file contains all the function prototypes for the <a href="#">gpio.c</a> file	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">i2c.h</a>	79
This file contains all the function prototypes for the <a href="#">i2c.c</a> file	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">INVERTER.h</a>	82
Header file for Finite State Machine (FSM) control	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">main.h</a>	86
: Header for <a href="#">main.c</a> file. This file contains the common defines of the application	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">MEASUREMENTS.h</a>	100
Header file for handling measurements	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">PCB_IO.h</a>	104
Header file for handling GPIOs and other low-priority tasks	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">Pergamon_float.h</a>	109
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">PWM.h</a>	114
Header file for controlling PWM output	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">stm32f7xx_hal_conf.h</a>	119
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">stm32f7xx_it.h</a>	138
This file contains the headers of the interrupt handlers	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">TASKS_1ms.h</a>	143
Header file for functions related to tasks executed every 1ms	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">tim.h</a>	145
This file contains all the function prototypes for the <a href="#">tim.c</a> file	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ <a href="#">usb_otg.h</a>	151
This file contains all the function prototypes for the <a href="#">usb_otg.c</a> file	

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">adc.c</a>	
This file provides code for the configuration of the ADC instances . . . . .	153
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">can.c</a>	
This file provides code for the configuration of the CAN instances . . . . .	158
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">CONTROL.c</a>	
Source file for control logic . . . . .	160
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">dac.c</a>	
This file provides code for the configuration of the DAC instances . . . . .	162
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">dma.c</a>	
This file provides code for the configuration of all the requested memory to memory DMA transfers . . . . .	165
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">FSM.c</a>	
This file provides code for Finite State Machine (FSM) control . . . . .	166
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">gpio.c</a>	
This file provides code for the configuration of all used GPIO pins . . . . .	168
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">I2c.c</a>	
This file provides code for the configuration of the I2C instances . . . . .	169
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">INVERTER.c</a>	
This file provides code for Finite State Machine (FSM) control . . . . .	171
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">main.c</a>	
: Main program body . . . . .	174
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">MEASUREMENTS.c</a>	
This file provides functions for handling measurements . . . . .	178
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">PCB_IO.c</a>	
This file provides functions for handling GPIOs and other low-priority tasks . . . . .	181
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">Pergamon_float.c</a>	
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">PWM.c</a>	
This file provides functions for controlling PWM output . . . . .	184
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">stm32f7xx_hal_msp.c</a>	
This file provides code for the MSP Initialization and de-Initialization codes . . . . .	188
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">stm32f7xx_it.c</a>	
Interrupt Service Routines . . . . .	189
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">syscalls.c</a>	
STM32CubeIDE Minimal System calls file . . . . .	194
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">sysmem.c</a>	
STM32CubeIDE System Memory calls file . . . . .	199
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">system_stm32f7xx.c</a>	
CMSIS Cortex-M7 Device Peripheral Access Layer System Source File . . . . .	200
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">TASKS_1ms.c</a>	
This file contains functions to execute tasks every 1ms . . . . .	202
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">tim.c</a>	
This file provides code for the configuration of the TIM instances . . . . .	203
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">usb_otg.c</a>	
This file provides code for the configuration of the USB_OTG instances . . . . .	210

# Chapter 4

## Topic Documentation

### 4.1 Math Constants

#### Macros

- #define **SQ2** 1.4142135624F
- #define **ISQ2** 0.7071067812F
- #define **SQ3** 1.7320508076F
- #define **ISQ3** 0.5773502692F
- #define **PI** 3.1415926536F
- #define **IPI** 0.3183098862F
- #define **PI2** 6.2831853072F
- #define **IPI2** 0.1591549431F
- #define **INV\_DEG** 0.0027777778F
- #define **INV3** 0.3333333333F
- #define **DIV2** 0.5F

#### 4.1.1 Detailed Description

#### 4.1.2 Macro Definition Documentation

##### 4.1.2.1 DIV2

```
#define DIV2 0.5F
```

1/2

##### 4.1.2.2 INV3

```
#define INV3 0.3333333333F
```

Inverse of 3

#### 4.1.2.3 INV\_DEG

```
#define INV_DEG 0.002777778F
```

Inverse of 360

#### 4.1.2.4 IPI

```
#define IPI 0.3183098862F
```

Inverse of Pi

#### 4.1.2.5 IPI2

```
#define IPI2 0.1591549431F
```

Inverse of (2\*Pi)

#### 4.1.2.6 ISQ2

```
#define ISQ2 0.7071067812F
```

Inverse of square root of 2

#### 4.1.2.7 ISQ3

```
#define ISQ3 0.5773502692F
```

Inverse of square root of 3

#### 4.1.2.8 PI

```
#define PI 3.1415926536F
```

Pi

#### 4.1.2.9 PI2

```
#define PI2 6.2831853072F
```

2\*Pi

#### 4.1.2.10 SQ2

```
#define SQ2 1.4142135624F
```

Square root of 2

### 4.1.2.11 SQ3

```
#define SQ3 1.7320508076F
```

Square root of 3

## 4.2 - Integral Controllers

### Data Structures

- struct [pi\\_aw\\_struct](#)  
*PI Controller with internal saturation, anti-windup, and feedforward.*
- struct [pi\\_struct](#)  
*PI Controller with external saturation and feedforward.*

### Functions

- void [pi\\_aw\\_calc](#) (volatile [pi\\_aw\\_struct](#) \*v) [\\_\\_attribute\\_\\_\(\(section\(".ccmram"\)\)\)](#)  
*Initializes the PI controller with anti-windup.*
- void [pi\\_init](#) (volatile [pi\\_struct](#) \*v) [\\_\\_attribute\\_\\_\(\(section\(".ccmram"\)\)\)](#)  
*Initializes the PI controller.*
- void [pi\\_calc](#) (volatile [pi\\_struct](#) \*v) [\\_\\_attribute\\_\\_\(\(section\(".ccmram"\)\)\)](#)  
*Calculates the output of the PI controller.*
- void [pi\\_extsat\\_calc](#) (volatile [pi\\_struct](#) \*v)  
*Calculates the output of the PI controller with external saturation.*

### 4.2.1 Detailed Description

### 4.2.2 Function Documentation

#### 4.2.2.1 [pi\\_aw\\_calc\(\)](#)

```
void pi_aw_calc (
    volatile pi\_aw\_struct * v )
```

Initializes the PI controller with anti-windup.

#### Parameters

v	Pointer to the PI controller structure.
---	---

Initializes the PI controller with anti-windup.

#### Parameters

v	Pointer to the PI control structure.
---	--------------------------------------

**Note**

This function computes the PI control action with anti-windup.

#### 4.2.2.2 pi\_calc()

```
void pi_calc (
    volatile pi_struct * v )
```

Calculates the output of the PI controller.

**Parameters**

v	Pointer to the PI controller structure.
---	---

Calculates the output of the PI controller.

**Parameters**

v	Pointer to the PI structure.
---	------------------------------

**Note**

This function computes the PI control action with feedforward and saturation.

#### 4.2.2.3 pi\_extsat\_calc()

```
void pi_extsat_calc (
    volatile pi_struct * v )
```

Calculates the output of the PI controller with external saturation.

**Parameters**

v	Pointer to the PI controller structure.
---	---

Calculates the output of the PI controller with external saturation.

**Parameters**

v	Pointer to the PI structure.
---	------------------------------

**Note**

This function computes the PI control action without saturation for external saturation.

#### 4.2.2.4 pi\_init()

```
void pi_init (
    volatile pi_struct * v )
```

Initializes the PI controller.

##### Parameters

v	Pointer to the PI controller structure.
---	---

Initializes the PI controller.

##### Parameters

v	Pointer to the PI structure.
---	------------------------------

##### Note

This function initializes the constants used in the PI controller.

## 4.3 Clarke and Park Transformations

### Data Structures

- struct `clarke3F_struct`  
*Clarke transformation for three-phase systems.*
- struct `iclarke3F_struct`  
*Inverse Clarke transformation for three-phase systems.*
- struct `rot_struct`  
*Rotates the DQ axis in the opposite direction (clockwise).*
- struct `irot_struct`  
*Inverse rotation (counterclockwise).*

### Functions

- void `clarke3F_calc` (volatile `clarke3F_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the Clarke transformation.*
- void `iclarke3F_calc` (volatile `iclarke3F_struct` \*v)  
*Calculates the inverse Clarke transformation.*
- void `rot_calc` (volatile `rot_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the rotation in the opposite direction.*
- void `irot_calc` (volatile `irot_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the inverse rotation.*

### 4.3.1 Detailed Description

### 4.3.2 Function Documentation

#### 4.3.2.1 clarke3F\_calc()

```
void clarke3F_calc (
    volatile clarke3F_struct * v )
```

Calculates the Clarke transformation.

**Parameters**

v	Pointer to the Clarke transformation structure.
---	---

Calculates the Clarke transformation.

**Parameters**

v	Pointer to the Clarke transformation structure.
---	---

**Note**

This function computes the Clarke transformation for three-phase signals.

#### 4.3.2.2 iclarke3F\_calc()

```
void iclarke3F_calc (
    volatile iclarke3F_struct * v )
```

Calculates the inverse Clarke transformation.

**Parameters**

v	Pointer to the inverse Clarke transformation structure.
---	---

Calculates the inverse Clarke transformation.

**Parameters**

v	Pointer to the inverse Clarke transformation structure.
---	---

**Note**

This function computes the inverse Clarke transformation for three-phase signals.

#### 4.3.2.3 irot\_calc()

```
void irot_calc (
    volatile irot_struct * v )
```

Calculates the inverse rotation.

**Parameters**

v	Pointer to the inverse rotation structure.
---	--

Calculates the inverse rotation.

**Parameters**

<code>v</code>	Pointer to the inverse rotation transformation structure.
----------------	---

**Note**

This function computes the inverse rotation transformation (counterclockwise).

**4.3.2.4 `rot_calc()`**

```
void rot_calc (
    volatile rot_struct * v )
```

Calculates the rotation in the opposite direction.

**Parameters**

<code>v</code>	Pointer to the rotation structure.
----------------	------------------------------------

Calculates the rotation in the opposite direction.

**Parameters**

<code>v</code>	Pointer to the rotation transformation structure.
----------------	---

**Note**

This function computes the rotation transformation (clockwise).

**4.4 Utility Functions****Data Structures**

- struct `angle_struct`  
*Generates an angle based on a fixed frequency.*
- struct `svpwm_struct`  
*Space Vector Pulse Width Modulation (SVPWM) implementation.*

**Functions**

- void `angle_calc` (volatile `angle_struct` \*p) `__attribute__((section(".ccmram")))`  
*Calculates the angle generation.*
- void `svpwm_calc` (volatile `svpwm_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the SVPWM outputs.*

#### 4.4.1 Detailed Description

#### 4.4.2 Function Documentation

##### 4.4.2.1 angle\_calc()

```
void angle_calc (
    volatile angle_struct * v )
```

Calculates the angle generation.

###### Parameters

<i>p</i>	Pointer to the angle structure.
----------	---------------------------------

Calculates the angle generation.

###### Parameters

<i>v</i>	Pointer to the angle generation structure.
----------	--

###### Note

This function generates the angle.

##### 4.4.2.2 svpwm\_calc()

```
void svpwm_calc (
    volatile svpwm_struct * v )
```

Calculates the SVPWM outputs.

###### Parameters

<i>v</i>	Pointer to the SVPWM structure.
----------	---------------------------------

Calculates the SVPWM outputs.

###### Parameters

<i>v</i>	Pointer to the SVPWM structure.
----------	---------------------------------

**Note**

This function calculates the Space Vector Pulse Width Modulation (SVPWM).

Here is the caller graph for this function:



## 4.5 Signal Processing Functions

### Data Structures

- struct `rampa_struct`  
*Single-ramp generator.*
- struct `rampa_dual_struct`  
*Dual-ramp generator.*
- struct `datalog_struct`
- struct `avg_struct_10`  
*Moving average filter for 10 samples.*
- struct `RMS_struct`  
*Root Mean Square (RMS) calculation.*
- struct `filtreLP_struct`  
*First-order low-pass filter.*

### Macros

- #define `N_DATALOG` 256  
*Data logger for logging variables.*

### Functions

- void `rampa_calc` (volatile `rampa_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the output of the single ramp generator.*
- void `rampa_dual_calc` (volatile `rampa_dual_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the output of the dual ramp generator.*
- void `datalog_calc` (volatile `datalog_struct` \*dl)  
*Calculates the data log.*
- void `avg_calc_10_samples` (volatile `avg_struct_10` \*v)  
*Calculates the moving average for 10 samples.*
- void `RMS_calc` (volatile `RMS_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the RMS.*
- void `filtreLP_init` (volatile `filtreLP_struct` \*v)  
*Initializes the first-order low-pass filter.*
- void `filtreLP_calc` (volatile `filtreLP_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the output of the first-order low-pass filter.*

### 4.5.1 Detailed Description

### 4.5.2 Macro Definition Documentation

#### 4.5.2.1 N\_DATALOG

```
#define N_DATALOG 256
```

Data logger for logging variables.

Number of samples for the data log

### 4.5.3 Function Documentation

#### 4.5.3.1 avg\_calc\_10\_samples()

```
void avg_calc_10_samples (
    volatile avg_struct_10 * v )
```

Calculates the moving average for 10 samples.

Parameters

v	Pointer to the moving average structure.
---	--

Calculates the moving average for 10 samples.

Parameters

v	Pointer to the structure containing the samples.
---	--

Note

This function calculates the average of 10 samples.

#### 4.5.3.2 datalog\_calc()

```
void datalog_calc (
    volatile datalog_struct * dl )
```

Calculates the data log.

Parameters

dl	Pointer to the data log structure.
dl	Pointer to the data log structure.

**Note**

This function calculates the data log.

#### 4.5.3.3 `filtreLP_calc()`

```
void filtreLP_calc (
    volatile filtreLP_struct * v )
```

Calculates the output of the first-order low-pass filter.

**Parameters**

<code>v</code>	Pointer to the filter structure.
----------------	----------------------------------

Calculates the output of the first-order low-pass filter.

**Parameters**

<code>v</code>	Pointer to the first-order filter structure.
----------------	--

**Note**

This function calculates the first-order filter.

#### 4.5.3.4 `filtreLP_init()`

```
void filtreLP_init (
    volatile filtreLP_struct * v )
```

Initializes the first-order low-pass filter.

**Parameters**

<code>v</code>	Pointer to the filter structure.
----------------	----------------------------------

Initializes the first-order low-pass filter.

**Parameters**

<code>v</code>	Pointer to the first-order filter structure.
----------------	--

**Note**

This function initializes the first-order filter.

#### 4.5.3.5 rampa\_calc()

```
void rampa_calc (
    volatile rampa_struct * v )
```

Calculates the output of the single ramp generator.

##### Parameters

v	Pointer to the single ramp generator structure.
---	---

Calculates the output of the single ramp generator.

##### Parameters

v	Pointer to the ramp structure.
---	--------------------------------

##### Note

This function calculates the ramp.

#### 4.5.3.6 rampa\_dual\_calc()

```
void rampa_dual_calc (
    volatile rampa_dual_struct * v )
```

Calculates the output of the dual ramp generator.

##### Parameters

v	Pointer to the dual ramp generator structure.
---	---

Calculates the output of the dual ramp generator.

##### Parameters

v	Pointer to the dual ramp structure.
---	-------------------------------------

##### Note

This function calculates the dual ramp.

#### 4.5.3.7 RMS\_calc()

```
void RMS_calc (
    volatile RMS_struct * v )
```

Calculates the RMS.

**Parameters**

v	Pointer to the RMS structure.
---	-------------------------------

Calculates the RMS.

**Parameters**

v	Pointer to the RMS structure.
---	-------------------------------

**Note**

This function calculates the Root Mean Square (RMS).

## 4.6 Miscellaneous Functions

### Data Structures

- struct [step\\_struct](#)

*Step function generator.*

### Functions

- void [step\\_calc](#) (volatile [step\\_struct](#) \*v) [\\_\\_attribute\\_\\_\(\(section\(".ccmram"\)\)\)](#)  
*Calculates the output of the step function generator.*

### 4.6.1 Detailed Description

### 4.6.2 Function Documentation

#### 4.6.2.1 [step\\_calc\(\)](#)

```
void step_calc (
    volatile step\_struct * v )
```

Calculates the output of the step function generator.

**Parameters**

v	Pointer to the step generator structure.
---	--

Calculates the output of the step function generator.

**Parameters**

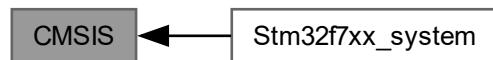
v	Pointer to the step structure.
---	--------------------------------

**Note**

This function calculates the step function.

## 4.7 CMSIS

Collaboration diagram for CMSIS:

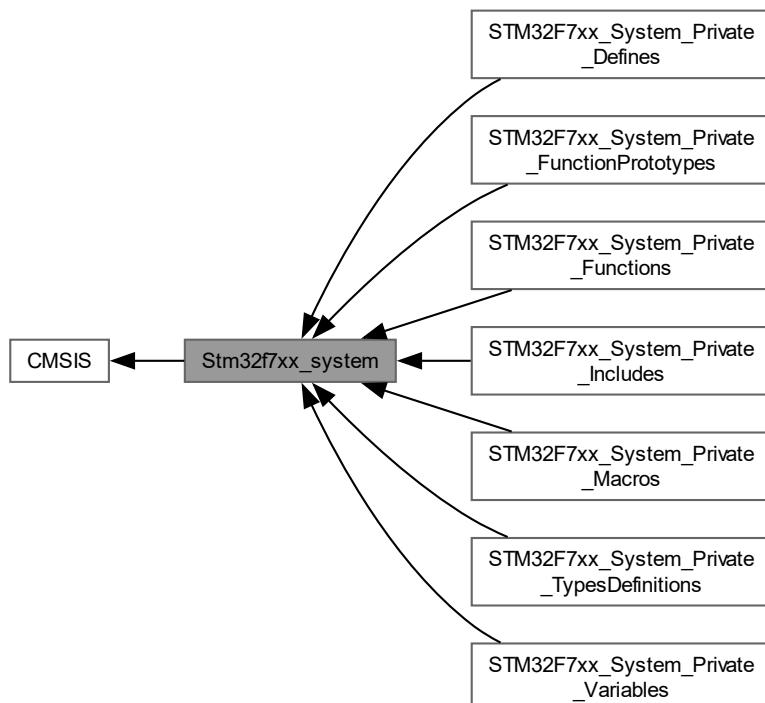
**Topics**

- [Stm32f7xx\\_system](#)

### 4.7.1 Detailed Description

### 4.7.2 Stm32f7xx\_system

Collaboration diagram for Stm32f7xx\_system:



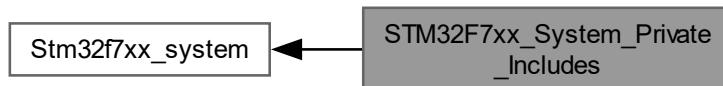
## Topics

- [STM32F7xx\\_System\\_Private\\_Includes](#)
- [STM32F7xx\\_System\\_Private\\_TypesDefinitions](#)
- [STM32F7xx\\_System\\_Private\\_Defines](#)
- [STM32F7xx\\_System\\_Private\\_Macros](#)
- [STM32F7xx\\_System\\_Private\\_Variables](#)
- [STM32F7xx\\_System\\_Private\\_FunctionPrototypes](#)
- [STM32F7xx\\_System\\_Private\\_Functions](#)

### 4.7.2.1 Detailed Description

### 4.7.2.2 STM32F7xx\_System\_Private\_Includes

Collaboration diagram for STM32F7xx\_System\_Private\_Includes:



## Macros

- [#define HSE\\_VALUE \(\(uint32\\_t\)25000000\)](#)
- [#define HSI\\_VALUE \(\(uint32\\_t\)16000000\)](#)

### 4.7.2.2.1 Detailed Description

### 4.7.2.2.2 Macro Definition Documentation

#### 4.7.2.2.2.1 HSE\_VALUE

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

Default value of the External oscillator in Hz

#### 4.7.2.2.2.2 HSI\_VALUE

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

Value of the Internal oscillator in Hz

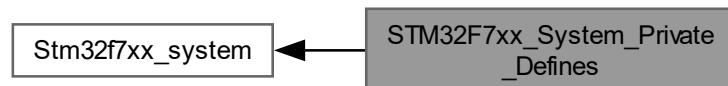
#### 4.7.2.3 STM32F7xx\_System\_Private\_TypesDefinitions

Collaboration diagram for STM32F7xx\_System\_Private\_TypesDefinitions:



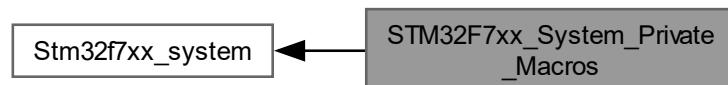
#### 4.7.2.4 STM32F7xx\_System\_Private\_Defines

Collaboration diagram for STM32F7xx\_System\_Private\_Defines:



#### 4.7.2.5 STM32F7xx\_System\_Private\_Macros

Collaboration diagram for STM32F7xx\_System\_Private\_Macros:



#### 4.7.2.6 STM32F7xx\_System\_Private\_Variables

Collaboration diagram for STM32F7xx\_System\_Private\_Variables:



#### Variables

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

##### 4.7.2.6.1 Detailed Description

##### 4.7.2.6.2 Variable Documentation

###### 4.7.2.6.2.1 AHBPrescTable

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

###### 4.7.2.6.2.2 APBPrescTable

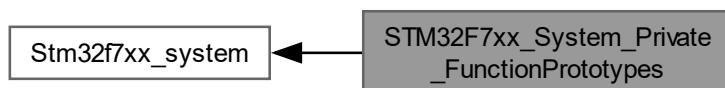
```
const uint8_t APBPrescTable[8] = {0, 0, 0, 0, 1, 2, 3, 4}
```

###### 4.7.2.6.2.3 SystemCoreClock

```
uint32_t SystemCoreClock = 16000000
```

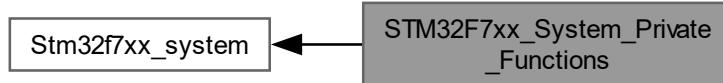
#### 4.7.2.7 STM32F7xx\_System\_Private\_FunctionPrototypes

Collaboration diagram for STM32F7xx\_System\_Private\_FunctionPrototypes:



#### 4.7.2.8 STM32F7xx\_System\_Private\_Functions

Collaboration diagram for STM32F7xx\_System\_Private\_Functions:



### Functions

- void [SystemInit](#) (void)  
*Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.*
- void [SystemCoreClockUpdate](#) (void)  
*Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.*

#### 4.7.2.8.1 Detailed Description

#### 4.7.2.8.2 Function Documentation

##### 4.7.2.8.2.1 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 [stm32f7xx\\_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 [stm32f7xx\\_hal\\_conf.h](#) file (default value 25 MHz), user has to ensure that HSE\_VALUE is same as the real frequency of the crystal used. Otherwise, this function may have wrong result.

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

**Parameters**

<i>None</i>	<input type="button" value=""/>
-------------	---------------------------------

**Return values**

<i>None</i>	<input type="button" value=""/>
-------------	---------------------------------

**4.7.2.8.2.2 SystemInit()**

```
void SystemInit (
    void )
```

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

**Parameters**

<i>None</i>	<input type="button" value=""/>
-------------	---------------------------------

**Return values**

<i>None</i>	<input type="button" value=""/>
-------------	---------------------------------

# Chapter 5

## Data Structure Documentation

### 5.1 angle\_struct Struct Reference

Generates an angle based on a fixed frequency.

```
#include <Pergamon_float.h>
```

#### Data Fields

- float freq
- float Ts
- float angle
- void(\* calc)()

#### 5.1.1 Detailed Description

Generates an angle based on a fixed frequency.

#### 5.1.2 Field Documentation

##### 5.1.2.1 angle

```
float angle
```

Network angle

##### 5.1.2.2 calc

```
void(* calc) ()
```

Pointer to the calculation function

### 5.1.2.3 freq

```
float freq
```

Frequency of the network

### 5.1.2.4 Ts

```
float Ts
```

Execution frequency of the angle integration

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.2 avg\_struct\_10 Struct Reference

Moving average filter for 10 samples.

```
#include <Pergamon_float.h>
```

### Data Fields

- float [out](#)
- float [in](#)[10]

### 5.2.1 Detailed Description

Moving average filter for 10 samples.

### 5.2.2 Field Documentation

#### 5.2.2.1 in

```
float in[10]
```

Input and past samples

#### 5.2.2.2 out

```
float out
```

Output variable

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.3 clarke3F\_struct Struct Reference

Clarke transformation for three-phase systems.

```
#include <Pergamon_float.h>
```

### Data Fields

- float **a**
- float **b**
- float **D**
- float **Q**
- void(\* **calc** )()

### 5.3.1 Detailed Description

Clarke transformation for three-phase systems.

### 5.3.2 Field Documentation

#### 5.3.2.1 **a**

```
float a
```

Phase A input

#### 5.3.2.2 **b**

```
float b
```

Phase B input

#### 5.3.2.3 **calc**

```
void(* calc) ()
```

Pointer to the calculation function

#### 5.3.2.4 **D**

```
float D
```

D-axis output

### 5.3.2.5 Q

```
float Q
```

Q-axis output

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.4 Compare3F\_struct Struct Reference

```
#include <PWM.h>
```

### Data Fields

- float [alfaA](#)
- float [alfaB](#)
- float [alfaC](#)
- uint32\_t [compA](#)
- uint32\_t [compB](#)
- uint32\_t [compC](#)
- void(\* [calc](#) )()

### 5.4.1 Field Documentation

#### 5.4.1.1 [alfaA](#)

```
float alfaA
```

#### 5.4.1.2 [alfaB](#)

```
float alfaB
```

#### 5.4.1.3 [alfaC](#)

```
float alfaC
```

#### 5.4.1.4 [calc](#)

```
void(* calc) ()
```

#### 5.4.1.5 compA

```
uint32_t compA
```

#### 5.4.1.6 compB

```
uint32_t compB
```

#### 5.4.1.7 compC

```
uint32_t compC
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[PWM.h](#)

## 5.5 **datalog\_struct** Struct Reference

```
#include <Pergamon_float.h>
```

### Data Fields

- `uint16_t i`
- `uint16_t j`
- `uint16_t estat`
- `uint16_t prescaler`
- `float * var`
- `void(* calc)()`
- `float log [N_DATALOG]`

### 5.5.1 Field Documentation

#### 5.5.1.1 calc

```
void(* calc) ()
```

Pointer to the calculation function

#### 5.5.1.2 estat

```
uint16_t estat
```

State: 0=stopped, 1=init, 2=running

### 5.5.1.3 i

uint16\_t i

State variable

### 5.5.1.4 j

uint16\_t j

State variable

### 5.5.1.5 log

float log[N\_DATALOG]

Log array

### 5.5.1.6 prescaler

uint16\_t prescaler

Prescaler

### 5.5.1.7 var

float\* var

Pointer to variables

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.6 Duties Struct Reference

Structure to hold PWM configuration parameters.

```
#include <PWM.h>
```

### Data Fields

- float [Da](#)
- float [Db](#)
- float [Dc](#)

### 5.6.1 Detailed Description

Structure to hold PWM configuration parameters.

### 5.6.2 Field Documentation

#### 5.6.2.1 Da

```
float Da
```

Duty cycle for channel 1

#### 5.6.2.2 Db

```
float Db
```

Duty cycle for channel 2

#### 5.6.2.3 Dc

```
float Dc
```

Duty cycle for channel 3

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[PWM.h](#)

## 5.7 Encoder Struct Reference

```
#include <MEASUREMENTS.h>
```

### Data Fields

- `uint16_t A`
- `uint16_t B`
- `uint16_t Z`
- `float wm_rpm`
- `float we`
- `float theta_m`
- `float theta_e`
- `uint8_t DIR`

## 5.7.1 Field Documentation

### 5.7.1.1 A

```
uint16_t A
```

Encoder channel A value

### 5.7.1.2 B

```
uint16_t B
```

Encoder channel B value

### 5.7.1.3 DIR

```
uint8_t DIR
```

Direction

### 5.7.1.4 theta\_e

```
float theta_e
```

Electrical rotor position

### 5.7.1.5 theta\_m

```
float theta_m
```

Mechanical rotor position

### 5.7.1.6 we

```
float we
```

Electrical angular velocity

### 5.7.1.7 wm\_rpm

```
float wm_rpm
```

Mechanical angular velocity (RPM)

### 5.7.1.8 Z

```
uint16_t Z
```

Encoder channel Z value

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[MEASUREMENTS.h](#)

## 5.8 **filtreLP\_struct** Struct Reference

First-order low-pass filter.

```
#include <Pergamon_float.h>
```

### Data Fields

- float **in**
- float **out**
- float **alfa**
- float **Ts**
- float **fc**
- uint16\_t **enable**
- void(\* **init**)()
- void(\* **calc**)()

### 5.8.1 Detailed Description

First-order low-pass filter.

### 5.8.2 Field Documentation

#### 5.8.2.1 alfa

```
float alfa
```

Filter coefficient

#### 5.8.2.2 calc

```
void(* calc) ()
```

Pointer to the calculation function

### 5.8.2.3 enable

```
uint16_t enable
```

Enable flag

### 5.8.2.4 fc

```
float fc
```

Cutoff frequency

### 5.8.2.5 in

```
float in
```

Input signal

### 5.8.2.6 init

```
void(* init) ()
```

Pointer to the initialization function

### 5.8.2.7 out

```
float out
```

Output signal

### 5.8.2.8 Ts

```
float Ts
```

Execution period

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.9 iclarke3F\_struct Struct Reference

Inverse Clarke transformation for three-phase systems.

```
#include <Pergamon_float.h>
```

## Data Fields

- float `D`
- float `Q`
- float `a`
- float `b`
- void(\* `calc` )()

### 5.9.1 Detailed Description

Inverse Clarke transformation for three-phase systems.

### 5.9.2 Field Documentation

#### 5.9.2.1 `a`

float `a`

Phase A output

#### 5.9.2.2 `b`

float `b`

Phase B output

#### 5.9.2.3 `calc`

void(\* `calc`) ()

Pointer to the calculation function

#### 5.9.2.4 `D`

float `D`

D-axis input

#### 5.9.2.5 `Q`

float `Q`

Q-axis input

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

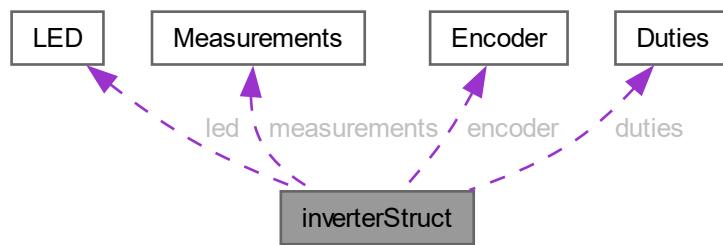
- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.10 inverterStruct Struct Reference

Inverter structure.

```
#include <INVERTER.h>
```

Collaboration diagram for inverterStruct:



### Data Fields

- `LED * led`
- `GPIO_TypeDef * enable_port`
- `uint16_t enable_pin`
- `TIM_HandleTypeDef * htim`
- `ADC_HandleTypeDef * hadc`
- `inverterState state`
- `uint32_t ADC_raw [4]`
- `Measurements measurements`
- `Encoder encoder`
- `Duties duties`

### 5.10.1 Detailed Description

Inverter structure.

### 5.10.2 Field Documentation

#### 5.10.2.1 ADC\_raw

```
uint32_t ADC_raw[4]
```

Raw ADC measurements

### 5.10.2.2 duties

`Duties` duties

Duty cycles for phase A,B,C

### 5.10.2.3 enable\_pin

`uint16_t enable_pin`

Pin number for enabling/disabling the inverter

### 5.10.2.4 enable\_port

`GPIO_TypeDef* enable_port`

GPIO port for enabling/disabling the inverter

### 5.10.2.5 encoder

`Encoder` encoder

Electrical and mechanical angles and speeds

### 5.10.2.6 hadc

`ADC_HandleTypeDef* hadc`

Handle of the ADC peripheral for the current phase currents and DC voltage sensing

### 5.10.2.7 htim

`TIM_HandleTypeDef* htim`

Handle of the timer peripheral for the PWM output

### 5.10.2.8 led

`LED*` led

`LED` control struct

### 5.10.2.9 measurements

`Measurements` measurements

Phase currents and DC voltage measurements

### 5.10.2.10 state

```
inverterState state
```

Current state of the inverter operation

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[INVERTER.h](#)

## 5.11 irot\_struct Struct Reference

Inverse rotation (counterclockwise).

```
#include <Pergamon_float.h>
```

### Data Fields

- float [d](#)
- float [q](#)
- float [sinFi](#)
- float [cosFi](#)
- float [D](#)
- float [Q](#)
- void(\* [calc](#))()

### 5.11.1 Detailed Description

Inverse rotation (counterclockwise).

### 5.11.2 Field Documentation

#### 5.11.2.1 calc

```
void(* calc) ()
```

Pointer to the calculation function

#### 5.11.2.2 cosFi

```
float cosFi
```

Cosine of the angle of rotation

**5.11.2.3 d**

```
float d
```

D-axis input

**5.11.2.4 D**

```
float D
```

D-axis output

**5.11.2.5 q**

```
float q
```

Q-axis input

**5.11.2.6 Q**

```
float Q
```

Q-axis output

**5.11.2.7 sinFi**

```
float sinFi
```

Sine of the angle of rotation

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.12 LED Struct Reference

[LED](#) structure.

```
#include <PCB_IO.h>
```

### Data Fields

- [GPIO\\_TypeDef \\* port](#)
- [uint16\\_t pin](#)
- [LEDMode mode](#)

### 5.12.1 Detailed Description

`LED` structure.

### 5.12.2 Field Documentation

#### 5.12.2.1 mode

`LEDMode mode`

Current `LED` mode

#### 5.12.2.2 pin

`uint16_t pin`

Pin number for controlling the `LED`

#### 5.12.2.3 port

`GPIO_TypeDef* port`

GPIO port for controlling the `LED`

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[PCB\\_IO.h](#)

## 5.13 Measurements Struct Reference

```
#include <MEASUREMENTS.h>
```

### Data Fields

- float `ia`
- float `ib`
- float `ic`
- float `VDC`

### 5.13.1 Field Documentation

#### 5.13.1.1 ia

`float ia`

Phase A current

### 5.13.1.2 ib

float ib

Phase B current

### 5.13.1.3 ic

float ic

Phase C current

### 5.13.1.4 VDC

float VDC

DC link voltage

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[MEASUREMENTS.h](#)

## 5.14 pi\_aw\_struct Struct Reference

PI Controller with internal saturation, anti-windup, and feedforward.

```
#include <Pergamon_float.h>
```

### Data Fields

- uint16\_t [enable](#)
- float [Ts](#)
- float [Kp](#)
- float [Ki](#)
- float [Kaw](#)
- float [e](#) [2]
- float [pi\\_consig](#)
- float [pi\\_fdb](#)
- float [pi\\_out\\_max](#)
- float [pi\\_out\\_min](#)
- float [pi\\_out\\_presat](#)
- float [pi\\_out\\_postsat](#)
- float [pi\\_out](#)
- float [pi\\_int](#) [2]
- float [pi\\_ffw](#) [2]
- void(\* [calc](#) )()

### 5.14.1 Detailed Description

PI Controller with internal saturation, anti-windup, and feedforward.

### 5.14.2 Field Documentation

#### 5.14.2.1 calc

```
void(* calc) ()
```

Pointer to the calculation function

#### 5.14.2.2 e

```
float e[2]
```

Error at current and previous step

#### 5.14.2.3 enable

```
uint16_t enable
```

Enable flag for the controller

#### 5.14.2.4 Kaw

```
float Kaw
```

Anti-windup gain

#### 5.14.2.5 Ki

```
float Ki
```

Integral gain

#### 5.14.2.6 Kp

```
float Kp
```

Proportional gain

**5.14.2.7 pi\_consig**

```
float pi_consig
```

Setpoint

**5.14.2.8 pi\_fdb**

```
float pi_fdb
```

Feedback

**5.14.2.9 pi\_ffw**

```
float pi_ffw[2]
```

Feedforward at current and previous step

**5.14.2.10 pi\_int**

```
float pi_int[2]
```

Integrator Part

**5.14.2.11 pi\_out**

```
float pi_out
```

Controller output

**5.14.2.12 pi\_out\_max**

```
float pi_out_max
```

Maximum output

**5.14.2.13 pi\_out\_min**

```
float pi_out_min
```

Minimum output

**5.14.2.14 pi\_out\_postsat**

```
float pi_out_postsat
```

Output after saturation

### 5.14.2.15 pi\_out\_presat

```
float pi_out_presat
```

Output before saturation

### 5.14.2.16 Ts

```
float Ts
```

Execution period

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.15 pi\_struct Struct Reference

PI Controller with external saturation and feedforward.

```
#include <Pergamon_float.h>
```

### Data Fields

- `uint16_t enable`
- `float Ts`
- `float Kp`
- `float Ki`
- `float K0`
- `float K1`
- `float e [2]`
- `float pi_consig`
- `float pi_fdb`
- `float pi_out_max`
- `float pi_out_min`
- `float pi_out`
- `float pi_ffw [2]`
- `void(* init)()`
- `void(* calc)()`

### 5.15.1 Detailed Description

PI Controller with external saturation and feedforward.

## 5.15.2 Field Documentation

### 5.15.2.1 calc

```
void(* calc) ()
```

Pointer to the calculation function

### 5.15.2.2 e

```
float e[2]
```

Error at current and previous step

### 5.15.2.3 enable

```
uint16_t enable
```

Enable flag for the controller

### 5.15.2.4 init

```
void(* init) ()
```

Pointer to the initialization function

### 5.15.2.5 K0

```
float K0
```

$$K0 = Kp + (Ts*Ki)/2$$

### 5.15.2.6 K1

```
float K1
```

$$K1 = -Kp + (Ts*Ki)/2$$

### 5.15.2.7 Ki

```
float Ki
```

Integral gain

**5.15.2.8 Kp**

float Kp

Proportional gain

**5.15.2.9 pi\_consig**

float pi\_consig

Setpoint

**5.15.2.10 pi\_fdb**

float pi\_fdb

Feedback

**5.15.2.11 pi\_ffw**

float pi\_ffw[2]

Feedforward at current and previous step

**5.15.2.12 pi\_out**

float pi\_out

Controller output

**5.15.2.13 pi\_out\_max**

float pi\_out\_max

Maximum output

**5.15.2.14 pi\_out\_min**

float pi\_out\_min

Minimum output

### 5.15.2.15 Ts

float Ts

Execution period

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon\_float.h

## 5.16 rampa\_dual\_struct Struct Reference

Dual-ramp generator.

```
#include <Pergamon_float.h>
```

### Data Fields

- float `in`
- float `out`
- float `Incr`
- float `Decr`
- uint8\_t `enable`
- void(\* `calc`)()

### 5.16.1 Detailed Description

Dual-ramp generator.

### 5.16.2 Field Documentation

#### 5.16.2.1 calc

```
void(* calc) ()
```

Pointer to the calculation function

#### 5.16.2.2 Decr

```
float Decr
```

Decrement

### 5.16.2.3 enable

uint8\_t enable

Enable flag

### 5.16.2.4 in

float in

Input signal

### 5.16.2.5 Incr

float Incr

Increment

### 5.16.2.6 out

float out

Output signal

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.17 rampa\_struct Struct Reference

Single-ramp generator.

```
#include <Pergamon_float.h>
```

### Data Fields

- float [in](#)
- float [out](#)
- float [Incr](#)
- uint8\_t [enable](#)
- void(\*) [calc](#) ()

### 5.17.1 Detailed Description

Single-ramp generator.

## 5.17.2 Field Documentation

### 5.17.2.1 calc

```
void(* calc) ()
```

Pointer to the calculation function

### 5.17.2.2 enable

```
uint8_t enable
```

Enable flag

### 5.17.2.3 in

```
float in
```

Input signal

### 5.17.2.4 incr

```
float incr
```

Increment

### 5.17.2.5 out

```
float out
```

Output signal

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.18 RMS\_struct Struct Reference

Root Mean Square (RMS) calculation.

```
#include <Pergamon_float.h>
```

## Data Fields

- float `T_exec`
- float `Measure`
- float `Sq_Sum`
- float `Out_RMS`
- float `Freq`
- float `Angle`
- float `Angle_ant`

### 5.18.1 Detailed Description

Root Mean Square (RMS) calculation.

### 5.18.2 Field Documentation

#### 5.18.2.1 Angle

float `Angle`

Angle

#### 5.18.2.2 Angle\_ant

float `Angle_ant`

Previous angle

#### 5.18.2.3 Freq

float `Freq`

Output frequency of the PLL

#### 5.18.2.4 Measure

float `Measure`

Signal to be RMSed

#### 5.18.2.5 Out\_RMS

float `Out_RMS`

RMSed signal

### 5.18.2.6 Sq\_Sum

float Sq\_Sum

Sum of squares

### 5.18.2.7 T\_exec

float T\_exec

Execution frequency of the function

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.19 rot\_struct Struct Reference

Rotates the DQ axis in the opposite direction (clockwise).

```
#include <Pergamon_float.h>
```

### Data Fields

- float **D**
- float **Q**
- float **sinFi**
- float **cosFi**
- float **d**
- float **q**
- void(\* **calc** )()

### 5.19.1 Detailed Description

Rotates the DQ axis in the opposite direction (clockwise).

### 5.19.2 Field Documentation

#### 5.19.2.1 calc

```
void(* calc) ()
```

Pointer to the calculation function

### 5.19.2.2 cosFi

float cosFi

Cosine of the angle of rotation

### 5.19.2.3 D

float D

D-axis input

### 5.19.2.4 d

float d

Rotated D-axis

### 5.19.2.5 Q

float Q

Q-axis input

### 5.19.2.6 q

float q

Rotated Q-axis

### 5.19.2.7 sinFi

float sinFi

Sine of the angle of rotation

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.20 step\_struct Struct Reference

Step function generator.

```
#include <Pergamon_float.h>
```

## Data Fields

- float `fs`
- float `In`
- float `Out`
- float `Step`
- float `t_step`
- uint32\_t `Pulses`
- uint32\_t `Counter`
- uint16\_t `enable`
- void(\* `calc` )()

### 5.20.1 Detailed Description

Step function generator.

### 5.20.2 Field Documentation

#### 5.20.2.1 `calc`

```
void(* calc) ()
```

Pointer to the calculation function

#### 5.20.2.2 `Counter`

```
uint32_t Counter
```

Counter for pulses

#### 5.20.2.3 `enable`

```
uint16_t enable
```

Enable flag

#### 5.20.2.4 `fs`

```
float fs
```

Function execution frequency

#### 5.20.2.5 `In`

```
float In
```

Input variable

### 5.20.2.6 Out

```
float Out
```

Output variable (with step when needed)

### 5.20.2.7 Pulses

```
uint32_t Pulses
```

Pulse counter for seconds

### 5.20.2.8 Step

```
float Step
```

Step amplitude

### 5.20.2.9 t\_step

```
float t_step
```

Step duration in seconds

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)

## 5.21 svpwm\_struct Struct Reference

Space Vector Pulse Width Modulation (SVPWM) implementation.

```
#include <Pergamon_float.h>
```

### Data Fields

- float [valpha](#)
- float [vbeta](#)
- float [Ta](#)
- float [Tb](#)
- float [Tc](#)
- void(\* [calc](#) )()

### 5.21.1 Detailed Description

Space Vector Pulse Width Modulation (SVPWM) implementation.

## 5.21.2 Field Documentation

### 5.21.2.1 calc

```
void(* calc) ()
```

Pointer to the calculation function

### 5.21.2.2 Ta

```
float Ta
```

Output for phase A (0-1)

### 5.21.2.3 Tb

```
float Tb
```

Output for phase B (0-1)

### 5.21.2.4 Tc

```
float Tc
```

Output for phase C (0-1)

### 5.21.2.5 valfa

```
float valfa
```

Input in per-unit for A phase

### 5.21.2.6 vbeta

```
float vbeta
```

Input in per-unit for B phase

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[Pergamon\\_float.h](#)



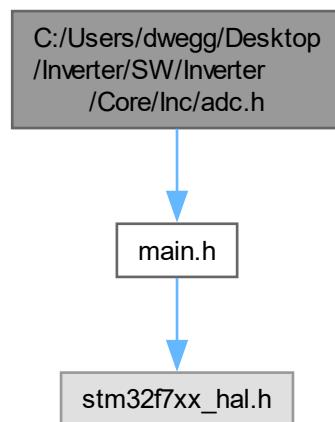
# Chapter 6

## File Documentation

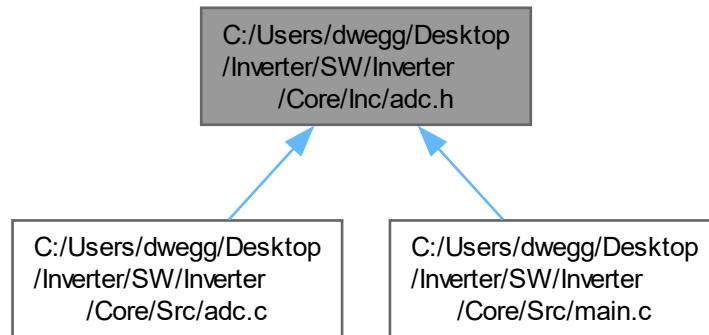
### 6.1 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/adc.h File Reference

This file contains all the function prototypes for the [adc.c](#) file.

```
#include "main.h"  
Include dependency graph for adc.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- void [MX\\_ADC1\\_Init](#) (void)
- void [MX\\_ADC2\\_Init](#) (void)
- void [MX\\_ADC3\\_Init](#) (void)

## Variables

- ADC\_HandleTypeDef [hadc1](#)
- ADC\_HandleTypeDef [hadc2](#)
- ADC\_HandleTypeDef [hadc3](#)

### 6.1.1 Detailed Description

This file contains all the function prototypes for the [adc.c](#) file.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

## 6.1.2 Function Documentation

### 6.1.2.1 MX\_ADC1\_Init()

```
void MX_ADC1_Init (
    void )
```

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

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

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

Configure for the selected ADC regular channel its corresponding rank in the sequencer and its sample time. Here is the call graph for this function:



Here is the caller graph for this function:



### 6.1.2.2 MX\_ADC2\_Init()

```
void MX_ADC2_Init (
    void )
```

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

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

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

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

Configure for the selected ADC regular channel its corresponding rank in the sequencer and its sample time. Here is the call graph for this function:



Here is the caller graph for this function:



### 6.1.2.3 MX\_ADC3\_Init()

```
void MX_ADC3_Init ( void )
```

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

Configure for the selected ADC regular channel its corresponding rank in the sequencer and its sample time. Here is the call graph for this function:



Here is the caller graph for this function:



### 6.1.3 Variable Documentation

#### 6.1.3.1 hadc1

```
ADC_HandleTypeDef hadc1 [extern]
```

#### 6.1.3.2 hadc2

```
ADC_HandleTypeDef hadc2 [extern]
```

#### 6.1.3.3 hadc3

```
ADC_HandleTypeDef hadc3 [extern]
```

## 6.2 adc.h

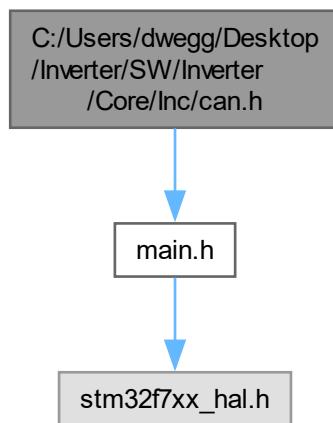
[Go to the documentation of this file.](#)

```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -----*/
00021 #ifndef __ADC_H__
00022 #define __ADC_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes -----*/
00029 #include "main.h"
00030
00031 /* USER CODE BEGIN Includes */
00032
00033 /* USER CODE END Includes */
00034
00035 extern ADC_HandleTypeDef hadc1;
00036
00037 extern ADC_HandleTypeDef hadc2;
00038
00039 extern ADC_HandleTypeDef hadc3;
00040
00041 /* USER CODE BEGIN Private defines */
00042
00043 /* USER CODE END Private defines */
00044
00045 void MX_ADC1_Init(void);
00046 void MX_ADC2_Init(void);
00047 void MX_ADC3_Init(void);
00048
00049 /* USER CODE BEGIN Prototypes */
00050
00051 /* USER CODE END Prototypes */
00052
00053 #ifdef __cplusplus
00054 }
00055 #endif
00056
00057 #endif /* __ADC_H__ */
00058
```

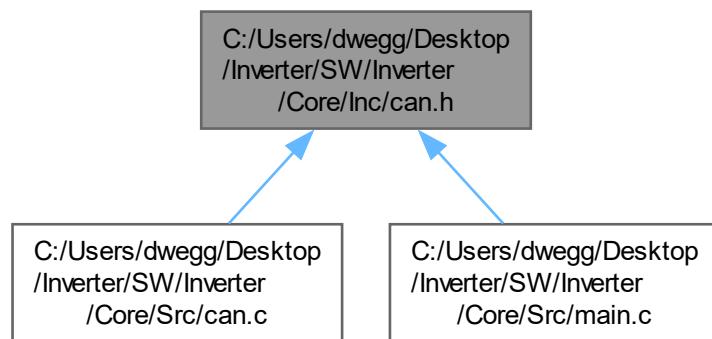
## 6.3 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/can.h File Reference

This file contains all the function prototypes for the [can.c](#) file.

```
#include "main.h"  
Include dependency graph for can.h:
```



This graph shows which files directly or indirectly include this file:



### Functions

- void [MX\\_CAN1\\_Init](#) (void)

## Variables

- CAN\_HandleTypeDef [hcan1](#)

### 6.3.1 Detailed Description

This file contains all the function prototypes for the [can.c](#) file.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

### 6.3.2 Function Documentation

#### 6.3.2.1 MX\_CAN1\_Init()

```
void MX_CAN1_Init (
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.3.3 Variable Documentation

#### 6.3.3.1 hcan1

```
CAN_HandleTypeDef hcan1 [extern]
```

## 6.4 can.h

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

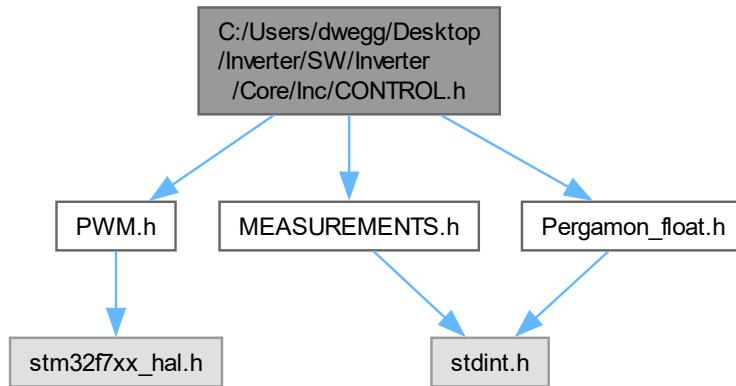
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -----*/
00021 #ifndef __CAN_H__
00022 #define __CAN_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes -----*/
00029 #include "main.h"
00030
00031 /* USER CODE BEGIN Includes */
00032
00033 /* USER CODE END Includes */
00034
00035 extern CAN_HandleTypeDef hcan1;
00036
00037 /* USER CODE BEGIN Private defines */
00038
00039 /* USER CODE END Private defines */
00040
00041 void MX_CAN1_Init(void);
00042
00043 /* USER CODE BEGIN Prototypes */
00044
00045 /* USER CODE END Prototypes */
00046
00047 #ifdef __cplusplus
00048 }
00049 #endif
00050
00051 #endif /* __CAN_H__ */
00052
```

## 6.5 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/CONTROL.h File Reference

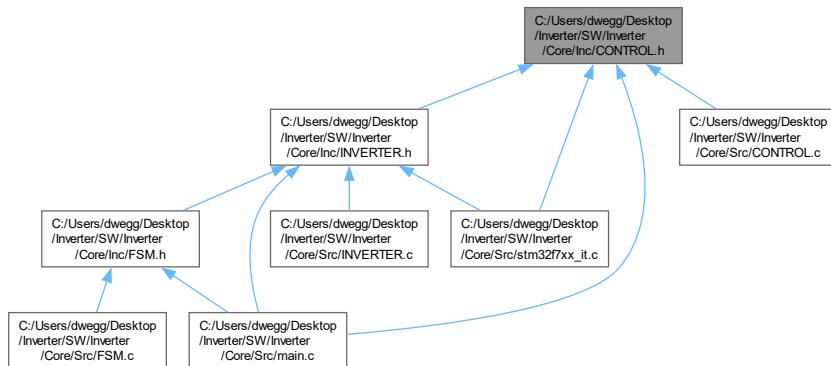
Header file for control logic.

```
#include "PWM.h"
#include "MEASUREMENTS.h"
```

```
#include "Pergamon_float.h"
Include dependency graph for CONTROL.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- void `control` (float `valfa`, float `vbeta`, Measurements `measurements`, volatile Duties \*`duties`)  
*Control function.*

## Variables

- float `valfa_L`
- float `vbeta_L`

### 6.5.1 Detailed Description

Header file for control logic.

## 6.5.2 Function Documentation

### 6.5.2.1 control()

```
void control (
    float valfa,
    float vbeta,
    Measurements measurements,
    volatile Duties * duties )
```

Control function.

This function implements the control logic based on measurements.

#### Parameters

<i>measurements</i>	Measurements structure.
<i>duties</i>	Pointer to the duties structure.

This function implements the control logic based on measurements. For now, it sets a constant duty cycle.

#### Parameters

<i>measurements</i>	Measurements structure.
<i>duties</i>	Pointer to the duties structure.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.5.3 Variable Documentation

#### 6.5.3.1 valfa\_L

```
float valfa_L [extern]
```

#### 6.5.3.2 vbeta\_L

```
float vbeta_L [extern]
```

## 6.6 CONTROL.h

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

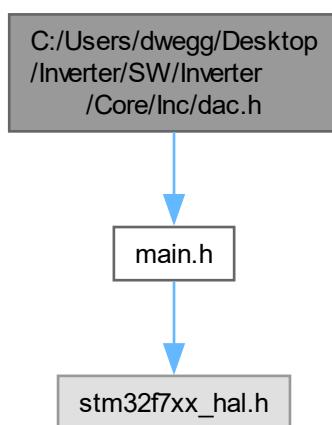
```
00001 #ifndef CONTROL_H
00002 #define CONTROL_H
00003
00004 #include "PWM.h"
00005 #include "MEASUREMENTS.h"
00006 #include "Pergamon_float.h"
00007
00013 extern float valfa_L;
00014 extern float vbeta_L;
00015
00016
00025 void control(float valfa, float vbeta, Measurements measurements, volatile Duties *duties);
00026
00027 #endif /* CONTROL_H */
```

## 6.7 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/dac.h File Reference

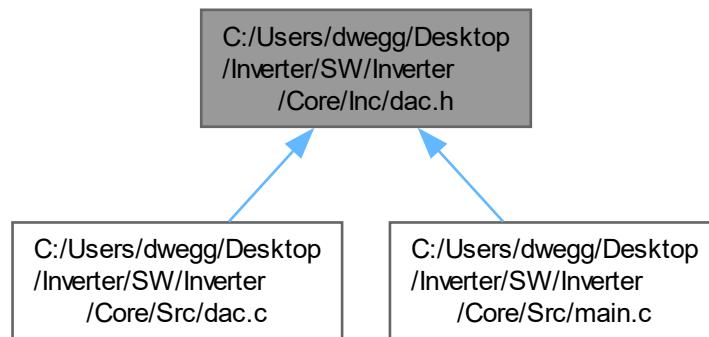
This file contains all the function prototypes for the [dac.c](#) file.

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

Include dependency graph for dac.h:



This graph shows which files directly or indirectly include this file:



## Functions

- void [MX\\_DAC\\_Init](#) (void)

## Variables

- DAC\_HandleTypeDef [hdac](#)

### 6.7.1 Detailed Description

This file contains all the function prototypes for the [dac.c](#) file.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

## 6.7.2 Function Documentation

### 6.7.2.1 MX\_DAC\_Init()

```
void MX_DAC_Init (
    void )
```

DAC Initialization

DAC channel OUT1 configHere is the call graph for this function:



Here is the caller graph for this function:



## 6.7.3 Variable Documentation

### 6.7.3.1 hdac

```
DAC_HandleTypeDef hdac [extern]
```

## 6.8 dac.h

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

```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -----*/
00021 #ifndef __DAC_H__
00022 #define __DAC_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes -----*/
00029 #include "main.h"
00030
```

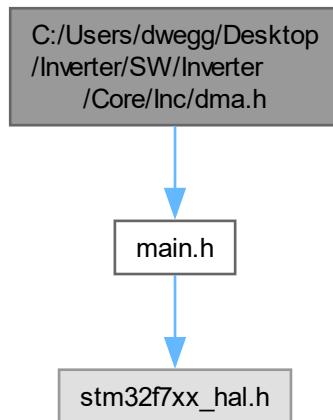
```
00031 /* USER CODE BEGIN Includes */
00032
00033 /* USER CODE END Includes */
00034
00035 extern DAC_HandleTypeDef hdac;
00036
00037 /* USER CODE BEGIN Private defines */
00038
00039 /* USER CODE END Private defines */
00040
00041 void MX_DAC_Init(void);
00042
00043 /* USER CODE BEGIN Prototypes */
00044
00045 /* USER CODE END Prototypes */
00046
00047 #ifdef __cplusplus
00048 }
00049 #endif
00050
00051 #endif /* __DAC_H__ */
00052
```

## 6.9 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/dma.h File Reference

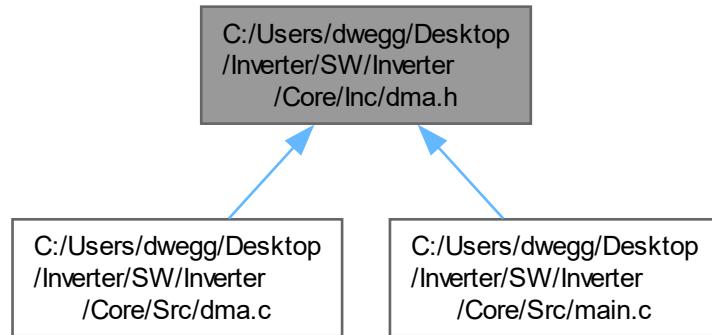
This file contains all the function prototypes for the [dma.c](#) file.

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

Include dependency graph for dma.h:



This graph shows which files directly or indirectly include this file:



## Functions

- void [MX\\_DMA\\_Init](#) (void)

### 6.9.1 Detailed Description

This file contains all the function prototypes for the [dma.c](#) file.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

### 6.9.2 Function Documentation

#### 6.9.2.1 MX\_DMA\_Init()

```
void MX_DMA_Init (
    void )
```

Enable DMA controller clock Here is the caller graph for this function:



## 6.10 dma.h

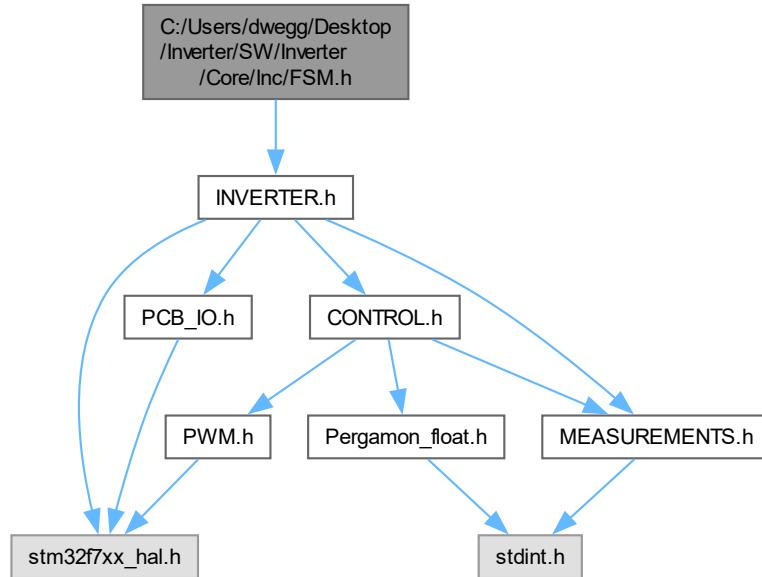
[Go to the documentation of this file.](#)

```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -----*/
00021 #ifndef __DMA_H__
00022 #define __DMA_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes -----*/
00029 #include "main.h"
00030
00031 /* DMA memory to memory transfer handles -----*/
00032
00033 /* USER CODE BEGIN Includes */
00034
00035 /* USER CODE END Includes */
00036
00037 /* USER CODE BEGIN Private defines */
00038
00039 /* USER CODE END Private defines */
00040
00041 void MX_DMA_Init(void);
00042
00043 /* USER CODE BEGIN Prototypes */
00044
00045 /* USER CODE END Prototypes */
00046
00047 #ifdef __cplusplus
00048 }
00049 #endif
00050
00051 #endif /* __DMA_H__ */
00052
```

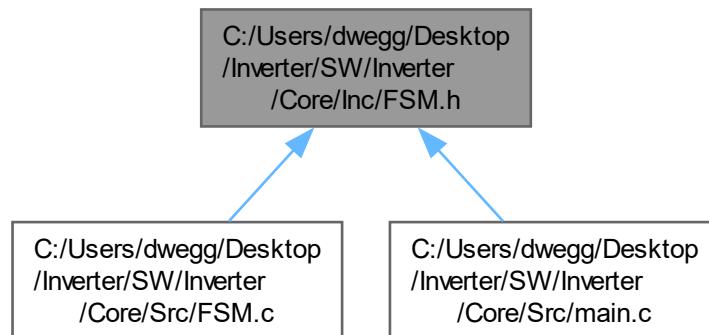
## 6.11 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/FSM.h File Reference

Header file for Finite State Machine (FSM) control.

```
#include "INVERTER.h"
Include dependency graph for FSM.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- void `inv_FSM (inverterStruct *inv)`  
*Run the Finite State Machine (FSM) for inverter operation control.*

### 6.11.1 Detailed Description

Header file for Finite State Machine (FSM) control.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg in GitHub). All rights reserved.

This software is licensed under the Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. For more information, see: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

### 6.11.2 Function Documentation

#### 6.11.2.1 inv\_FSM()

```
void inv_FSM (
    inverterStruct * inv )
```

Run the Finite State Machine (FSM) for inverter operation control.

##### Parameters

<i>inv</i>	Pointer to the inverter structure.
------------	------------------------------------

Run the Finite State Machine (FSM) for inverter operation control.

This function executes the finite state machine to control the inverter based on its current state.

##### Parameters

<i>inv</i>	Pointer to the inverter structure.
------------	------------------------------------

Here is the caller graph for this function:



## 6.12 FSM.h

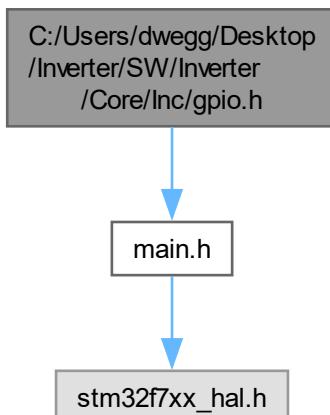
[Go to the documentation of this file.](#)

```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020 #ifndef FSM_H
00021 #define FSM_H
00022
00023 #include "INVERTER.h"
00024
00030 void inv_FSM(inverterStruct *inv);
00031
00032 #endif /* FSM_H */
```

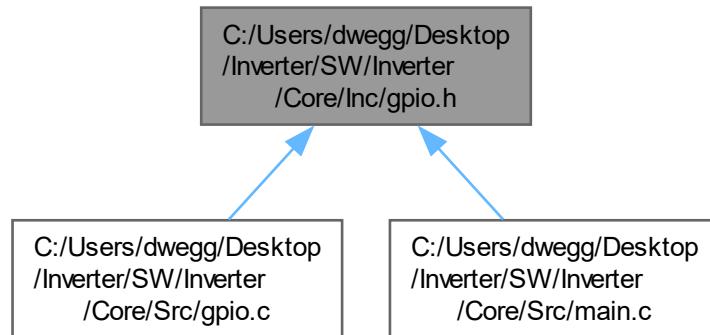
## 6.13 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/gpio.h File Reference

This file contains all the function prototypes for the [gpio.c](#) file.

```
#include "main.h"
Include dependency graph for gpio.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- void [MX\\_GPIO\\_Init](#) (void)

### 6.13.1 Detailed Description

This file contains all the function prototypes for the [gpio.c](#) file.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

### 6.13.2 Function Documentation

#### 6.13.2.1 MX\_GPIO\_Init()

```
void MX_GPIO_Init (
    void )
```

Configure pins as Analog Input Output EVENT\_OUT EXTI PA9 -----> USB\_OTG\_FS\_VBUS PA10 -----> USB\_<\_OTG\_FS\_ID PA11 -----> USB\_OTG\_FS\_DM PA12 -----> USB\_OTG\_FS\_DP Here is the caller graph for this function:



## 6.14 gpio.h

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

```

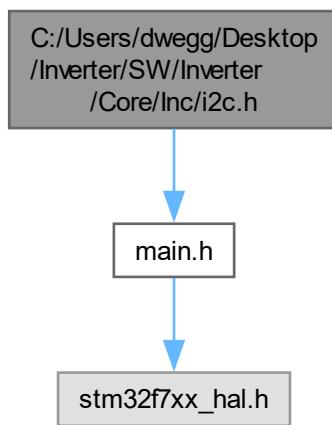
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -----*/
00021 #ifndef __GPIO_H__
00022 #define __GPIO_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes -----*/
00029 #include "main.h"
00030
00031 /* USER CODE BEGIN Includes */
00032
00033 /* USER CODE END Includes */
00034
00035 /* USER CODE BEGIN Private defines */
00036
00037 /* USER CODE END Private defines */
00038
00039 void MX_GPIO_Init(void);
00040
00041 /* USER CODE BEGIN Prototypes */
00042
00043 /* USER CODE END Prototypes */
00044
00045 #ifdef __cplusplus
00046 }
00047 #endif
00048 #endif /*__ GPIO_H__ */
00049

```

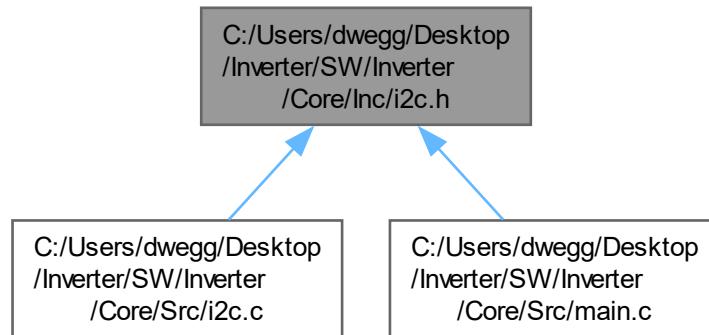
## 6.15 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/i2c.h File Reference

This file contains all the function prototypes for the [i2c.c](#) file.

```
#include "main.h"
Include dependency graph for i2c.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- void [MX\\_I2C1\\_Init](#) (void)

## Variables

- I2C\_HandleTypeDef [hi2c1](#)

### 6.15.1 Detailed Description

This file contains all the function prototypes for the [i2c.c](#) file.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

## 6.15.2 Function Documentation

### 6.15.2.1 MX\_I2C1\_Init()

```
void MX_I2C1_Init (
    void )
```

Configure Analogue filter

Configure Digital filterHere is the call graph for this function:



Here is the caller graph for this function:



## 6.15.3 Variable Documentation

### 6.15.3.1 hi2c1

```
I2C_HandleTypeDef hi2c1 [extern]
```

## 6.16 i2c.h

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

```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -----*/
00021 #ifndef __I2C_H__
00022 #define __I2C_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes -----*/
00029 #include "main.h"
00030
```

```

00031 /* USER CODE BEGIN Includes */
00032
00033 /* USER CODE END Includes */
00034
00035 extern I2C_HandleTypeDef hi2c1;
00036
00037 /* USER CODE BEGIN Private defines */
00038
00039 /* USER CODE END Private defines */
00040
00041 void MX_I2C1_Init(void);
00042
00043 /* USER CODE BEGIN Prototypes */
00044
00045 /* USER CODE END Prototypes */
00046
00047 #ifdef __cplusplus
00048 }
00049 #endif
00050
00051 #endif /* __I2C_H__ */
00052

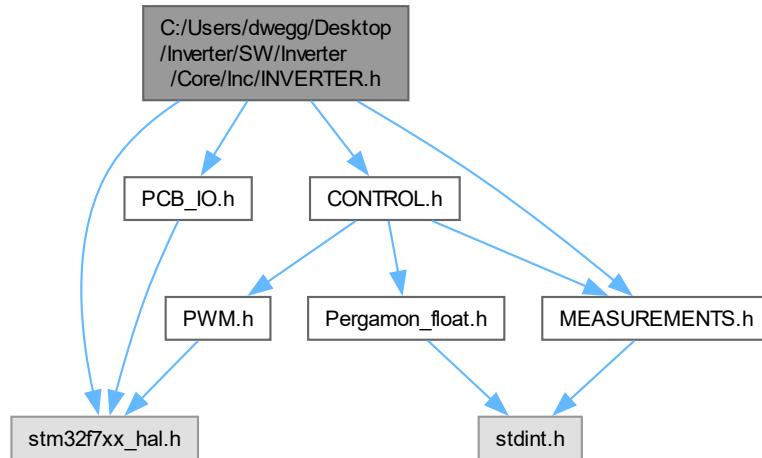
```

## 6.17 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/INVERTER.h File Reference

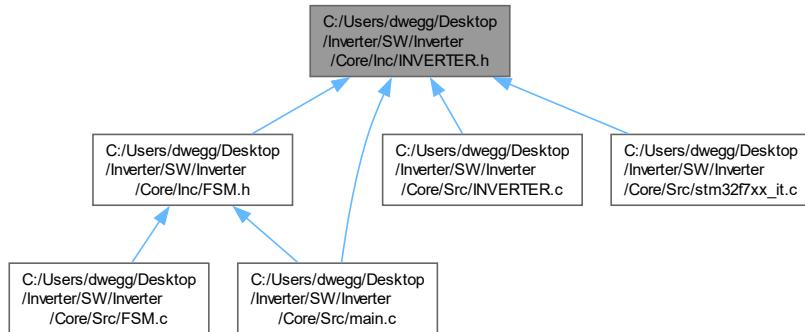
Header file for Finite State Machine (FSM) control.

```
#include "stm32f7xx_hal.h"
#include "PCB_IO.h"
#include "MEASUREMENTS.h"
#include "CONTROL.h"
```

Include dependency graph for INVERTER.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct `inverterStruct`  
*Inverter structure.*

## Enumerations

- enum `inverterState` { `INV_STATE_IDLE` , `INV_STATE_STARTUP` , `INV_STATE_RUNNING` , `INV_STATE_FAULT` }  
*Enumeration of inverter operation states.*

## Functions

- void `inv_init` (`inverterStruct` \*`inv`, `LED` \*`led`, `GPIO_TypeDef` \*`enable_port`, `uint16_t` `enable_pin`, `TIM_HandleTypeDef` \*`htim`, `ADC_HandleTypeDef` \*`hadc`)  
*Initialize the inverter.*

## Variables

- `inverterStruct invLeft`
- `inverterStruct invRight`

### 6.17.1 Detailed Description

Header file for Finite State Machine (FSM) control.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg in GitHub). All rights reserved.

This software is licensed under the Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. For more information, see: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

## 6.17.2 Enumeration Type Documentation

### 6.17.2.1 inverterState

```
enum inverterState
```

Enumeration of inverter operation states.

#### Enumerator

INV_STATE_IDLE	Inverter idle state
INV_STATE_STARTUP	Inverter startup state
INV_STATE_RUNNING	Inverter running state
INV_STATE_FAULT	Inverter fault state

## 6.17.3 Function Documentation

### 6.17.3.1 inv\_init()

```
void inv_init (
    inverterStruct * inv,
    LED * led,
    GPIO_TypeDef * enable_port,
    uint16_t enable_pin,
    TIM_HandleTypeDef * htim,
    ADC_HandleTypeDef * hadc )
```

Initialize the inverter.

This function initializes the inverter structure with the specified [LED](#), GPIO port, and pin.

#### Parameters

<i>inv</i>	Pointer to the inverter structure.
<i>led</i>	Pointer to the <a href="#">LED</a> structure.
<i>enable_port</i>	Pointer to the GPIO port for enabling/disabling the inverter.
<i>enable_pin</i>	Pin number for enabling/disabling the inverter.
<i>htim</i>	Timer peripheral for the PWM output.
<i>hadc</i>	ADC peripheral for the current phase current and DC voltage sensing.

Here is the call graph for this function:



Here is the caller graph for this function:



## 6.17.4 Variable Documentation

### 6.17.4.1 invLeft

```
inverterStruct invLeft [extern]
```

### 6.17.4.2 invRight

```
inverterStruct invRight [extern]
```

## 6.18 INVERTER.h

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

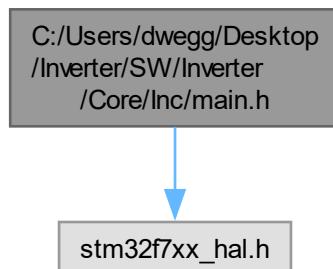
```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020 #ifndef INVERTER_H
00021 #define INVERTER_H
00022
00023 #include "stm32f7xx_hal.h"
00024 #include "PCB_IO.h"
00025 #include "MEASUREMENTS.h"
00026 #include "CONTROL.h"
00027
00031 typedef enum {
00032     INV_STATE_IDLE,
00033     INV_STATE_STARTUP,
00034     INV_STATE_RUNNING,
00035     INV_STATE_FAULT
00036 } inverterState;
00037
00038
```

```
00042 typedef struct {
00043     LED *led;
00044     GPIO_TypeDef *enable_port;
00045     uint16_t enable_pin;
00046     TIM_HandleTypeDef *htim;
00047     ADC_HandleTypeDef *hadc;
00048     inverterState state;
00049     uint32_t ADC_raw[4];
00050     Measurements measurements;
00051     Encoder encoder;
00052     Duties duties;
00053 } inverterStruct;
00054
00055
00056 extern inverterStruct invLeft;
00057 extern inverterStruct invRight;
00058
00072 void inv_init(inverterStruct *inv, LED *led, GPIO_TypeDef *enable_port, uint16_t enable_pin,
00073                 TIM_HandleTypeDef *htim, ADC_HandleTypeDef *hadc);
00074
00075 #endif /* INVERTER_H */
```

## 6.19 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/main.h File Reference

: Header for `main.c` file. This file contains the common defines of the application.

```
#include "stm32f7xx_hal.h"  
Include dependency graph for main.h:
```



This graph shows which files directly or indirectly include this file:



## Macros

- #define TS 0.00002

- #define DT 0.00000015
- #define Tinv\_L\_Pin GPIO\_PIN\_0
- #define Tinv\_L\_GPIO\_Port GPIOC
- #define Tinv\_R\_Pin GPIO\_PIN\_1
- #define Tinv\_R\_GPIO\_Port GPIOC
- #define Tmot\_L\_Pin GPIO\_PIN\_2
- #define Tmot\_L\_GPIO\_Port GPIOC
- #define Tmot\_R\_Pin GPIO\_PIN\_3
- #define Tmot\_R\_GPIO\_Port GPIOC
- #define ia\_L\_Pin GPIO\_PIN\_0
- #define ia\_L\_GPIO\_Port GPIOA
- #define ib\_L\_Pin GPIO\_PIN\_1
- #define ib\_L\_GPIO\_Port GPIOA
- #define ic\_L\_Pin GPIO\_PIN\_2
- #define ic\_L\_GPIO\_Port GPIOA
- #define VDC\_L\_Pin GPIO\_PIN\_3
- #define VDC\_L\_GPIO\_Port GPIOA
- #define DAC\_Pin GPIO\_PIN\_4
- #define DAC\_GPIO\_Port GPIOA
- #define PWM1\_R\_Pin GPIO\_PIN\_5
- #define PWM1\_R\_GPIO\_Port GPIOA
- #define ia\_R\_Pin GPIO\_PIN\_6
- #define ia\_R\_GPIO\_Port GPIOA
- #define ib\_R\_Pin GPIO\_PIN\_7
- #define ib\_R\_GPIO\_Port GPIOA
- #define SC\_det\_Pin GPIO\_PIN\_4
- #define SC\_det\_GPIO\_Port GPIOC
- #define ic\_R\_Pin GPIO\_PIN\_0
- #define ic\_R\_GPIO\_Port GPIOB
- #define VDC\_R\_Pin GPIO\_PIN\_1
- #define VDC\_R\_GPIO\_Port GPIOB
- #define ENABLE\_R\_Pin GPIO\_PIN\_2
- #define ENABLE\_R\_GPIO\_Port GPIOB
- #define ENABLE\_L\_Pin GPIO\_PIN\_7
- #define ENABLE\_L\_GPIO\_Port GPIOE
- #define PWM1\_L\_Pin GPIO\_PIN\_8
- #define PWM1\_L\_GPIO\_Port GPIOE
- #define PWM2\_L\_Pin GPIO\_PIN\_9
- #define PWM2\_L\_GPIO\_Port GPIOE
- #define PWM3\_L\_Pin GPIO\_PIN\_10
- #define PWM3\_L\_GPIO\_Port GPIOE
- #define PWM4\_L\_Pin GPIO\_PIN\_11
- #define PWM4\_L\_GPIO\_Port GPIOE
- #define PWM5\_L\_Pin GPIO\_PIN\_12
- #define PWM5\_L\_GPIO\_Port GPIOE
- #define PWM6\_L\_Pin GPIO\_PIN\_13
- #define PWM6\_L\_GPIO\_Port GPIOE
- #define WRN\_L\_Pin GPIO\_PIN\_14
- #define WRN\_L\_GPIO\_Port GPIOE
- #define WRN\_R\_Pin GPIO\_PIN\_15
- #define WRN\_R\_GPIO\_Port GPIOE
- #define B\_R\_Pin GPIO\_PIN\_10
- #define B\_R\_GPIO\_Port GPIOB
- #define Z\_R\_Pin GPIO\_PIN\_11
- #define Z\_R\_GPIO\_Port GPIOB

- #define PWM3\_R\_Pin GPIO\_PIN\_14
- #define PWM3\_R\_GPIO\_Port GPIOB
- #define PWM5\_R\_Pin GPIO\_PIN\_15
- #define PWM5\_R\_GPIO\_Port GPIOB
- #define A\_L\_Pin GPIO\_PIN\_12
- #define A\_L\_GPIO\_Port GPIOD
- #define B\_L\_Pin GPIO\_PIN\_14
- #define B\_L\_GPIO\_Port GPIOD
- #define Z\_L\_Pin GPIO\_PIN\_15
- #define Z\_L\_GPIO\_Port GPIOD
- #define PWM2\_R\_Pin GPIO\_PIN\_6
- #define PWM2\_R\_GPIO\_Port GPIOC
- #define PWM4\_R\_Pin GPIO\_PIN\_7
- #define PWM4\_R\_GPIO\_Port GPIOC
- #define PWM6\_R\_Pin GPIO\_PIN\_8
- #define PWM6\_R\_GPIO\_Port GPIOC
- #define TRIP\_R\_Pin GPIO\_PIN\_9
- #define TRIP\_R\_GPIO\_Port GPIOC
- #define TRIP\_L\_Pin GPIO\_PIN\_8
- #define TRIP\_L\_GPIO\_Port GPIOA
- #define A\_R\_Pin GPIO\_PIN\_15
- #define A\_R\_GPIO\_Port GPIOA
- #define DIR\_Pin GPIO\_PIN\_3
- #define DIR\_GPIO\_Port GPIOD
- #define LED\_LEFT\_Pin GPIO\_PIN\_4
- #define LED\_LEFT\_GPIO\_Port GPIOD
- #define LED\_RIGHT\_Pin GPIO\_PIN\_5
- #define LED\_RIGHT\_GPIO\_Port GPIOD
- #define LED\_ERR\_Pin GPIO\_PIN\_6
- #define LED\_ERR\_GPIO\_Port GPIOD

## Functions

- void Error\_Handler (void)  
*This function is executed in case of error occurrence.*

### 6.19.1 Detailed Description

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

#### Attention

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

## 6.19.2 Macro Definition Documentation

### 6.19.2.1 A\_L\_GPIO\_Port

```
#define A_L_GPIO_Port GPIOD
```

### 6.19.2.2 A\_L\_Pin

```
#define A_L_Pin GPIO_PIN_12
```

### 6.19.2.3 A\_R\_GPIO\_Port

```
#define A_R_GPIO_Port GPIOA
```

### 6.19.2.4 A\_R\_Pin

```
#define A_R_Pin GPIO_PIN_15
```

### 6.19.2.5 B\_L\_GPIO\_Port

```
#define B_L_GPIO_Port GPIOD
```

### 6.19.2.6 B\_L\_Pin

```
#define B_L_Pin GPIO_PIN_14
```

### 6.19.2.7 B\_R\_GPIO\_Port

```
#define B_R_GPIO_Port GPIOB
```

### 6.19.2.8 B\_R\_Pin

```
#define B_R_Pin GPIO_PIN_10
```

### 6.19.2.9 DAC\_GPIO\_Port

```
#define DAC_GPIO_Port GPIOA
```

### 6.19.2.10 DAC\_Pin

```
#define DAC_Pin GPIO_PIN_4
```

**6.19.2.11 DIR\_GPIO\_Port**

```
#define DIR_GPIO_Port GPIOD
```

**6.19.2.12 DIR\_Pin**

```
#define DIR_Pin GPIO_PIN_3
```

**6.19.2.13 DT**

```
#define DT 0.00000015
```

**6.19.2.14 ENABLE\_L\_GPIO\_Port**

```
#define ENABLE_L_GPIO_Port GPIOE
```

**6.19.2.15 ENABLE\_L\_Pin**

```
#define ENABLE_L_Pin GPIO_PIN_7
```

**6.19.2.16 ENABLE\_R\_GPIO\_Port**

```
#define ENABLE_R_GPIO_Port GPIOB
```

**6.19.2.17 ENABLE\_R\_Pin**

```
#define ENABLE_R_Pin GPIO_PIN_2
```

**6.19.2.18 ia\_L\_GPIO\_Port**

```
#define ia_L_GPIO_Port GPIOA
```

**6.19.2.19 ia\_L\_Pin**

```
#define ia_L_Pin GPIO_PIN_0
```

**6.19.2.20 ia\_R\_GPIO\_Port**

```
#define ia_R_GPIO_Port GPIOA
```

**6.19.2.21 ia\_R\_Pin**

```
#define ia_R_Pin GPIO_PIN_6
```

**6.19.2.22 ib\_L\_GPIO\_Port**

```
#define ib_L_GPIO_Port GPIOA
```

**6.19.2.23 ib\_L\_Pin**

```
#define ib_L_Pin GPIO_PIN_1
```

**6.19.2.24 ib\_R\_GPIO\_Port**

```
#define ib_R_GPIO_Port GPIOA
```

**6.19.2.25 ib\_R\_Pin**

```
#define ib_R_Pin GPIO_PIN_7
```

**6.19.2.26 ic\_L\_GPIO\_Port**

```
#define ic_L_GPIO_Port GPIOA
```

**6.19.2.27 ic\_L\_Pin**

```
#define ic_L_Pin GPIO_PIN_2
```

**6.19.2.28 ic\_R\_GPIO\_Port**

```
#define ic_R_GPIO_Port GPIOB
```

**6.19.2.29 ic\_R\_Pin**

```
#define ic_R_Pin GPIO_PIN_0
```

**6.19.2.30 LED\_ERR\_GPIO\_Port**

```
#define LED_ERR_GPIO_Port GPIOD
```

**6.19.2.31 LED\_ERR\_Pin**

```
#define LED_ERR_Pin GPIO_PIN_6
```

**6.19.2.32 LED\_LEFT\_GPIO\_Port**

```
#define LED_LEFT_GPIO_Port GPIOD
```

**6.19.2.33 LED\_LEFT\_Pin**

```
#define LED_LEFT_Pin GPIO_PIN_4
```

**6.19.2.34 LED\_RIGHT\_GPIO\_Port**

```
#define LED_RIGHT_GPIO_Port GPIOD
```

**6.19.2.35 LED\_RIGHT\_Pin**

```
#define LED_RIGHT_Pin GPIO_PIN_5
```

**6.19.2.36 PWM1\_L\_GPIO\_Port**

```
#define PWM1_L_GPIO_Port GPIOE
```

**6.19.2.37 PWM1\_L\_Pin**

```
#define PWM1_L_Pin GPIO_PIN_8
```

**6.19.2.38 PWM1\_R\_GPIO\_Port**

```
#define PWM1_R_GPIO_Port GPIOA
```

**6.19.2.39 PWM1\_R\_Pin**

```
#define PWM1_R_Pin GPIO_PIN_5
```

**6.19.2.40 PWM2\_L\_GPIO\_Port**

```
#define PWM2_L_GPIO_Port GPIOE
```

**6.19.2.41 PWM2\_L\_Pin**

```
#define PWM2_L_Pin GPIO_PIN_9
```

**6.19.2.42 PWM2\_R\_GPIO\_Port**

```
#define PWM2_R_GPIO_Port GPIOC
```

**6.19.2.43 PWM2\_R\_Pin**

```
#define PWM2_R_Pin GPIO_PIN_6
```

**6.19.2.44 PWM3\_L\_GPIO\_Port**

```
#define PWM3_L_GPIO_Port GPIOE
```

**6.19.2.45 PWM3\_L\_Pin**

```
#define PWM3_L_Pin GPIO_PIN_10
```

**6.19.2.46 PWM3\_R\_GPIO\_Port**

```
#define PWM3_R_GPIO_Port GPIOB
```

**6.19.2.47 PWM3\_R\_Pin**

```
#define PWM3_R_Pin GPIO_PIN_14
```

**6.19.2.48 PWM4\_L\_GPIO\_Port**

```
#define PWM4_L_GPIO_Port GPIOE
```

**6.19.2.49 PWM4\_L\_Pin**

```
#define PWM4_L_Pin GPIO_PIN_11
```

**6.19.2.50 PWM4\_R\_GPIO\_Port**

```
#define PWM4_R_GPIO_Port GPIOC
```

**6.19.2.51 PWM4\_R\_Pin**

```
#define PWM4_R_Pin GPIO_PIN_7
```

**6.19.2.52 PWM5\_L\_GPIO\_Port**

```
#define PWM5_L_GPIO_Port GPIOE
```

**6.19.2.53 PWM5\_L\_Pin**

```
#define PWM5_L_Pin GPIO_PIN_12
```

**6.19.2.54 PWM5\_R\_GPIO\_Port**

```
#define PWM5_R_GPIO_Port GPIOB
```

**6.19.2.55 PWM5\_R\_Pin**

```
#define PWM5_R_Pin GPIO_PIN_15
```

**6.19.2.56 PWM6\_L\_GPIO\_Port**

```
#define PWM6_L_GPIO_Port GPIOE
```

**6.19.2.57 PWM6\_L\_Pin**

```
#define PWM6_L_Pin GPIO_PIN_13
```

**6.19.2.58 PWM6\_R\_GPIO\_Port**

```
#define PWM6_R_GPIO_Port GPIOC
```

**6.19.2.59 PWM6\_R\_Pin**

```
#define PWM6_R_Pin GPIO_PIN_8
```

**6.19.2.60 SC\_det\_GPIO\_Port**

```
#define SC_det_GPIO_Port GPIOC
```

### 6.19.2.61 SC\_det\_Pin

```
#define SC_det_Pin GPIO_PIN_4
```

### 6.19.2.62 Tinv\_L\_GPIO\_Port

```
#define Tinv_L_GPIO_Port GPIOC
```

### 6.19.2.63 Tinv\_L\_Pin

```
#define Tinv_L_Pin GPIO_PIN_0
```

### 6.19.2.64 Tinv\_R\_GPIO\_Port

```
#define Tinv_R_GPIO_Port GPIOC
```

### 6.19.2.65 Tinv\_R\_Pin

```
#define Tinv_R_Pin GPIO_PIN_1
```

### 6.19.2.66 Tmot\_L\_GPIO\_Port

```
#define Tmot_L_GPIO_Port GPIOC
```

### 6.19.2.67 Tmot\_L\_Pin

```
#define Tmot_L_Pin GPIO_PIN_2
```

### 6.19.2.68 Tmot\_R\_GPIO\_Port

```
#define Tmot_R_GPIO_Port GPIOC
```

### 6.19.2.69 Tmot\_R\_Pin

```
#define Tmot_R_Pin GPIO_PIN_3
```

### 6.19.2.70 TRIP\_L\_GPIO\_Port

```
#define TRIP_L_GPIO_Port GPIOA
```

**6.19.2.71 TRIP\_L\_Pin**

```
#define TRIP_L_Pin GPIO_PIN_8
```

**6.19.2.72 TRIP\_R\_GPIO\_Port**

```
#define TRIP_R_GPIO_Port GPIOC
```

**6.19.2.73 TRIP\_R\_Pin**

```
#define TRIP_R_Pin GPIO_PIN_9
```

**6.19.2.74 TS**

```
#define TS 0.00002
```

**6.19.2.75 VDC\_L\_GPIO\_Port**

```
#define VDC_L_GPIO_Port GPIOA
```

**6.19.2.76 VDC\_L\_Pin**

```
#define VDC_L_Pin GPIO_PIN_3
```

**6.19.2.77 VDC\_R\_GPIO\_Port**

```
#define VDC_R_GPIO_Port GPIOB
```

**6.19.2.78 VDC\_R\_Pin**

```
#define VDC_R_Pin GPIO_PIN_1
```

**6.19.2.79 WRN\_L\_GPIO\_Port**

```
#define WRN_L_GPIO_Port GPIOE
```

**6.19.2.80 WRN\_L\_Pin**

```
#define WRN_L_Pin GPIO_PIN_14
```

### 6.19.2.81 WRN\_R\_GPIO\_Port

```
#define WRN_R_GPIO_Port GPIOE
```

### 6.19.2.82 WRN\_R\_Pin

```
#define WRN_R_Pin GPIO_PIN_15
```

### 6.19.2.83 Z\_L\_GPIO\_Port

```
#define Z_L_GPIO_Port GPIOD
```

### 6.19.2.84 Z\_L\_Pin

```
#define Z_L_Pin GPIO_PIN_15
```

### 6.19.2.85 Z\_R\_GPIO\_Port

```
#define Z_R_GPIO_Port GPIOB
```

### 6.19.2.86 Z\_R\_Pin

```
#define Z_R_Pin GPIO_PIN_11
```

## 6.19.3 Function Documentation

### 6.19.3.1 Error\_Handler()

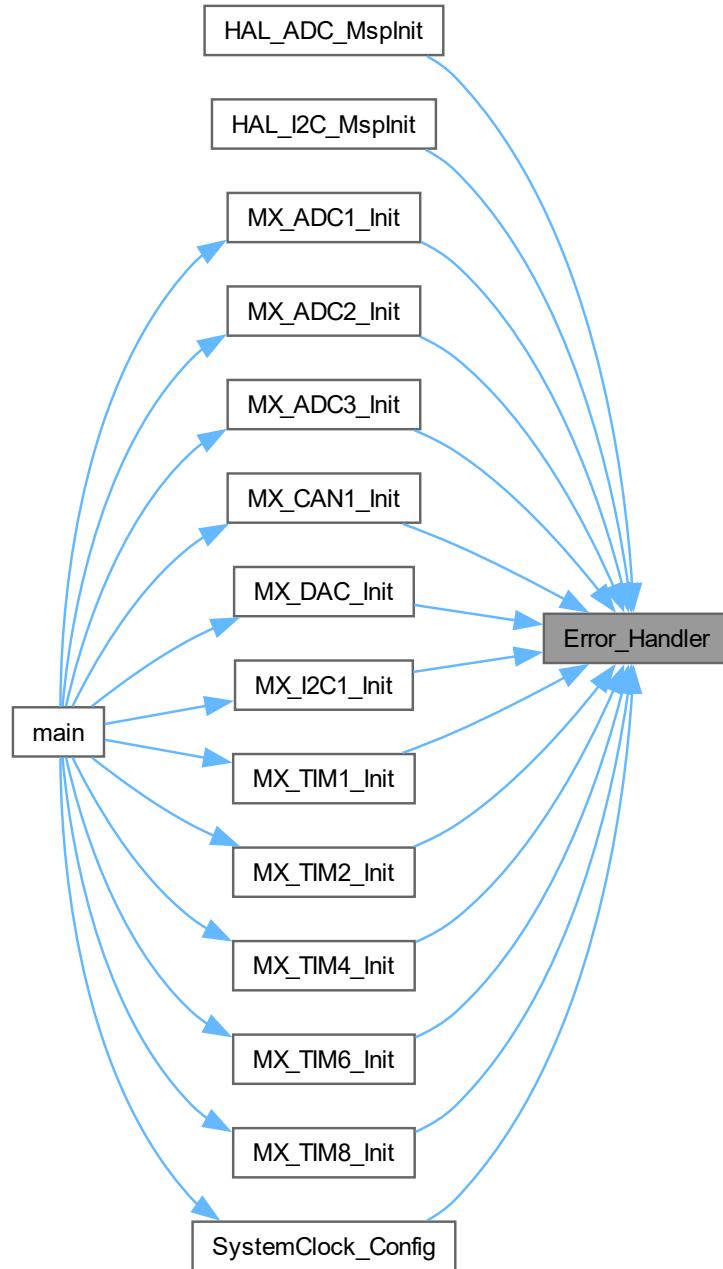
```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

Return values

None	<input type="button" value=""/>
------	---------------------------------

Here is the caller graph for this function:



## 6.20 main.h

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

```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020
00021 /* Define to prevent recursive inclusion -----*/
```

```
00022 #ifndef __MAIN_H
00023 #define __MAIN_H
00024
00025 #ifdef __cplusplus
00026 extern "C" {
00027 #endif
00028
00029 /* Includes -----*/
00030 #include "stm32f7xx_hal.h"
00031
00032 /* Private includes -----*/
00033 /* USER CODE BEGIN Includes */
00034
00035 /* USER CODE END Includes */
00036
00037 /* Exported types -----*/
00038 /* USER CODE BEGIN ET */
00039
00040 /* USER CODE END ET */
00041
00042 /* Exported constants -----*/
00043 /* USER CODE BEGIN EC */
00044
00045 /* USER CODE END EC */
00046
00047 /* Exported macro -----*/
00048 /* USER CODE BEGIN EM */
00049
00050 /* USER CODE END EM */
00051
00052 /* Exported functions prototypes -----*/
00053 void Error_Handler(void);
00054
00055 /* USER CODE BEGIN EFP */
00056
00057 /* USER CODE END EFP */
00058
00059 /* Private defines -----*/
00060 #define TS 0.00002
00061 #define DT 0.00000015
00062 #define Tinv_L_Pin GPIO_PIN_0
00063 #define Tinv_L_GPIO_Port GPIOC
00064 #define Tinv_R_Pin GPIO_PIN_1
00065 #define Tinv_R_GPIO_Port GPIOC
00066 #define Tmot_L_Pin GPIO_PIN_2
00067 #define Tmot_L_GPIO_Port GPIOC
00068 #define Tmot_R_Pin GPIO_PIN_3
00069 #define Tmot_R_GPIO_Port GPIOC
00070 #define ia_L_Pin GPIO_PIN_0
00071 #define ia_L_GPIO_Port GPIOA
00072 #define ib_L_Pin GPIO_PIN_1
00073 #define ib_L_GPIO_Port GPIOA
00074 #define ic_L_Pin GPIO_PIN_2
00075 #define ic_L_GPIO_Port GPIOA
00076 #define VDC_L_Pin GPIO_PIN_3
00077 #define VDC_L_GPIO_Port GPIOA
00078 #define DAC_Pin GPIO_PIN_4
00079 #define DAC_GPIO_Port GPIOA
00080 #define PWM1_R_Pin GPIO_PIN_5
00081 #define PWM1_R_GPIO_Port GPIOA
00082 #define ia_R_Pin GPIO_PIN_6
00083 #define ia_R_GPIO_Port GPIOA
00084 #define ib_R_Pin GPIO_PIN_7
00085 #define ib_R_GPIO_Port GPIOA
00086 #define SC_det_Pin GPIO_PIN_4
00087 #define SC_det_GPIO_Port GPIOC
00088 #define ic_R_Pin GPIO_PIN_0
00089 #define ic_R_GPIO_Port GPIOB
00090 #define VDC_R_Pin GPIO_PIN_1
00091 #define VDC_R_GPIO_Port GPIOB
00092 #define ENABLE_R_Pin GPIO_PIN_2
00093 #define ENABLE_R_GPIO_Port GPIOB
00094 #define ENABLE_L_Pin GPIO_PIN_7
00095 #define ENABLE_L_GPIO_Port GPIOE
00096 #define PWM1_L_Pin GPIO_PIN_8
00097 #define PWM1_L_GPIO_Port GPIOE
00098 #define PWM2_L_Pin GPIO_PIN_9
00099 #define PWM2_L_GPIO_Port GPIOE
00100 #define PWM3_L_Pin GPIO_PIN_10
00101 #define PWM3_L_GPIO_Port GPIOE
00102 #define PWM4_L_Pin GPIO_PIN_11
00103 #define PWM4_L_GPIO_Port GPIOE
00104 #define PWM5_L_Pin GPIO_PIN_12
00105 #define PWM5_L_GPIO_Port GPIOE
00106 #define PWM6_L_Pin GPIO_PIN_13
00107 #define PWM6_L_GPIO_Port GPIOE
00108 #define WRN_L_Pin GPIO_PIN_14
```

```

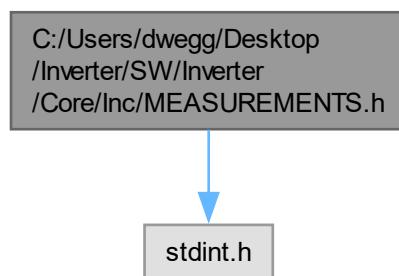
00109 #define WRN_L_GPIO_Port GPIOE
00110 #define WRN_R_Pin GPIO_PIN_15
00111 #define WRN_R_GPIO_Port GPIOE
00112 #define B_R_Pin GPIO_PIN_10
00113 #define B_R_GPIO_Port GPIOB
00114 #define Z_R_Pin GPIO_PIN_11
00115 #define Z_R_GPIO_Port GPIOB
00116 #define PWM3_R_Pin GPIO_PIN_14
00117 #define PWM3_R_GPIO_Port GPIOB
00118 #define PWM5_R_Pin GPIO_PIN_15
00119 #define PWM5_R_GPIO_Port GPIOB
00120 #define A_L_Pin GPIO_PIN_12
00121 #define A_L_GPIO_Port GPIOD
00122 #define B_L_Pin GPIO_PIN_14
00123 #define B_L_GPIO_Port GPIOD
00124 #define Z_L_Pin GPIO_PIN_15
00125 #define Z_L_GPIO_Port GPIOD
00126 #define PWM2_R_Pin GPIO_PIN_6
00127 #define PWM2_R_GPIO_Port GPIOC
00128 #define PWM4_R_Pin GPIO_PIN_7
00129 #define PWM4_R_GPIO_Port GPIOC
00130 #define PWM6_R_Pin GPIO_PIN_8
00131 #define PWM6_R_GPIO_Port GPIOC
00132 #define TRIP_R_Pin GPIO_PIN_9
00133 #define TRIP_R_GPIO_Port GPIOC
00134 #define TRIP_L_Pin GPIO_PIN_8
00135 #define TRIP_L_GPIO_Port GPIOA
00136 #define A_R_Pin GPIO_PIN_15
00137 #define A_R_GPIO_Port GPIOA
00138 #define DIR_Pin GPIO_PIN_3
00139 #define DIR_GPIO_Port GPIOD
00140 #define LED_LEFT_Pin GPIO_PIN_4
00141 #define LED_LEFT_GPIO_Port GPIOD
00142 #define LED_RIGHT_Pin GPIO_PIN_5
00143 #define LED_RIGHT_GPIO_Port GPIOD
00144 #define LED_ERR_Pin GPIO_PIN_6
00145 #define LED_ERR_GPIO_Port GPIOD
00146
00147 /* USER CODE BEGIN Private defines */
00148
00149 /* USER CODE END Private defines */
00150
00151 #ifdef __cplusplus
00152 }
00153 #endif
00154
00155 #endif /* __MAIN_H */

```

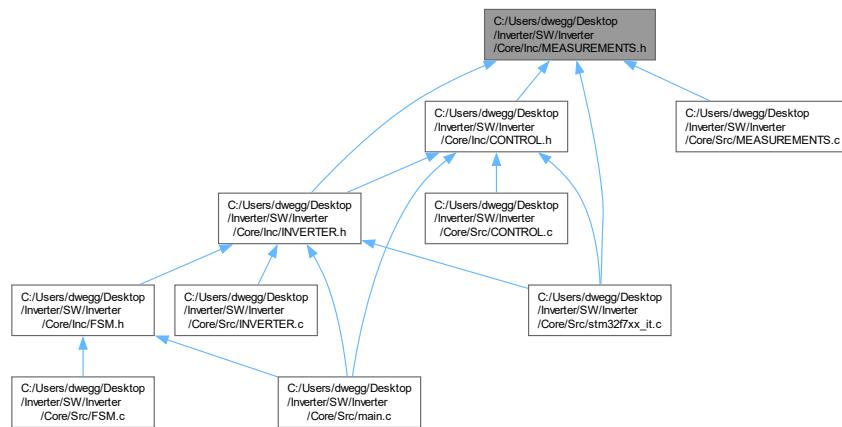
## 6.21 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/← MEASUREMENTS.h File Reference

Header file for handling measurements.

```
#include <stdint.h>
Include dependency graph for MEASUREMENTS.h:
```



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct [Encoder](#)
- struct [Measurements](#)

## Macros

- #define CURRENT\_SLOPE 54.4217687f
- #define CURRENT\_OFFSET 1.70068027211f
- #define VOLTAGE\_SLOPE 0.003796f
- #define VOLTAGE\_OFFSET 0.02083f

## Functions

- uint8\_t [getADCElec](#) (volatile uint32\_t \*ADC\_raw, volatile [Measurements](#) \*measurements)  
*Get electrical ADC measurements.*
- float [getLinear](#) (uint32\_t bits, float slope, float offset)  
*Convert ADC reading to physical measurement with linear response.*

### 6.21.1 Detailed Description

Header file for handling measurements.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg in GitHub). All rights reserved.

This software is licensed under the Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. For more information, see: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

## 6.21.2 Macro Definition Documentation

### 6.21.2.1 CURRENT\_OFFSET

```
#define CURRENT_OFFSET 1.70068027211f
```

[V]  $(10/(4.7+10)) * 2.5$  V

### 6.21.2.2 CURRENT\_SLOPE

```
#define CURRENT_SLOPE 54.4217687f
```

[A/V]  $(10/(4.7+10)) * (1 / (12.5 \text{ mV} / \text{A}))$

### 6.21.2.3 VOLTAGE\_OFFSET

```
#define VOLTAGE_OFFSET 0.02083f
```

[V]  $(100/(4700+100) * 5$  V

### 6.21.2.4 VOLTAGE\_SLOPE

```
#define VOLTAGE_SLOPE 0.003796f
```

[V/V]  $1/3 * 0.011388$  V

## 6.21.3 Function Documentation

### 6.21.3.1 getADCelec()

```
uint8_t getADCelec (
    volatile uint32_t * ADC_raw,
    volatile Measurements * measurements )
```

Get electrical ADC measurements.

#### Parameters

<i>ADC_raw</i>	Pointer to the raw ADC values array.
<i>measurements</i>	Pointer to the measurements struct to store the results.

#### Return values

<i>OK</i>	0 if an error occurred, 1 if successful.
-----------	--

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.21.3.2 getLinear()

```
float getLinear (
    uint32_t bits,
    float slope,
    float offset )
```

Convert ADC reading to physical measurement with linear response.

#### Parameters

<i>bits</i>	The ADC reading.
<i>slope</i>	The slope (volts per unit).
<i>offset</i>	The offset (volts at zero).

#### Return values

<i>measurement</i>	The physical measurement.
--------------------	---------------------------

#### Parameters

<i>bits</i>	The ADC reading.
<i>slope</i>	The slope (units per volt).
<i>offset</i>	The offset (volts at zero).

## Return values

<i>measurement</i>	The physical measurement.
--------------------	---------------------------

Here is the caller graph for this function:



## 6.22 MEASUREMENTS.h

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

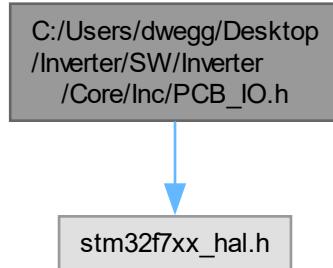
```

00001 /* USER CODE BEGIN Header */
00017 /* USER CODE END Header */
00018
00019 /* Define current and voltage gains/offsets */
00020 #define CURRENT_SLOPE 54.4217687f
00021 #define CURRENT_OFFSET 1.70068027211f
00022 #define VOLTAGE_SLOPE 0.003796f
00023 #define VOLTAGE_OFFSET 0.02083f
00026 #ifndef MEASUREMENTS_H
00027 #define MEASUREMENTS_H
00028
00029 /* Include necessary header files */
00030 #include <stdint.h>
00031
00032 /* Define encoder struct */
00033 typedef struct {
00034     uint16_t A;
00035     uint16_t B;
00036     uint16_t Z;
00037     float wm_rpm;
00038     float we;
00039     float theta_m;
00040     float theta_e;
00041     uint8_t DIR;
00042 } Encoder;
00043
00044 /* Define measurements struct */
00045 typedef struct {
00046     float ia;
00047     float ib;
00048     float ic;
00049     float VDC;
00050 } Measurements;
00051
00052 /* Define function prototypes */
00059 uint8_t getADCelec(volatile uint32_t* ADC_raw, volatile Measurements* measurements);
00067 float getLinear(uint32_t bits, float slope, float offset);
00068
00069 #endif /* MEASUREMENTS_H */
  
```

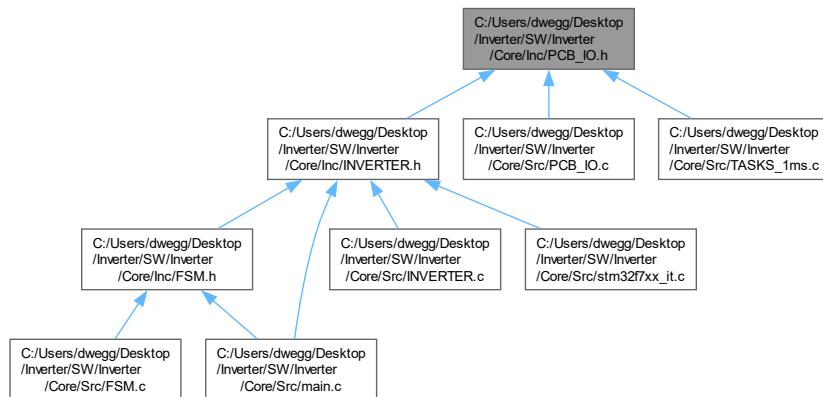
## 6.23 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PCB\_IO.h File Reference

Header file for handling GPIOs and other low-priority tasks.

```
#include "stm32f7xx_hal.h"
Include dependency graph for PCB_IO.h:
```



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct [LED](#)

*LED structure.*

## Macros

- #define [SC\\_DET\\_STATE\(\)](#) (HAL\_GPIO\_ReadPin(SC\_det\_GPIO\_Port, SC\_det\_Pin))
- #define [DIR\\_STATE\(\)](#) (HAL\_GPIO\_ReadPin(DIR\_GPIO\_Port, DIR\_Pin))
- #define [WRN\\_STATE](#)(port, pin) (HAL\_GPIO\_ReadPin(port, pin))
- #define [ENABLE](#)(port, pin) do { HAL\_GPIO\_WritePin(port, pin, GPIO\_PIN\_SET); } while(0)
- #define [DISABLE](#)(port, pin) do { HAL\_GPIO\_WritePin(port, pin, GPIO\_PIN\_RESET); } while(0)

## Enumerations

- enum LEDMode { LED\_MODE\_BLINK\_FAST , LED\_MODE\_BLINK\_SLOW , LED\_MODE\_ON , LED\_MODE\_OFF }

## Functions

- void LED\_handler (LED \*led, uint32\_t ms\_counter)  
*LED handler function.*

## Variables

- LED led\_left
- LED led\_right
- LED led\_error

### 6.23.1 Detailed Description

Header file for handling GPIOs and other low-priority tasks.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg in GitHub). All rights reserved.

This software is licensed under the Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. For more information, see: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

### 6.23.2 Macro Definition Documentation

#### 6.23.2.1 DIR\_STATE

```
#define DIR_STATE( ) (HAL_GPIO_ReadPin(DIR_GPIO_Port, DIR_Pin))
```

#### 6.23.2.2 DISABLE

```
#define DISABLE(
    port,
    pin ) do { HAL_GPIO_WritePin(port, pin, GPIO_PIN_RESET); } while(0)
```

#### 6.23.2.3 ENABLE

```
#define ENABLE(
    port,
    pin ) do { HAL_GPIO_WritePin(port, pin, GPIO_PIN_SET); } while(0)
```

#### 6.23.2.4 SC\_DET\_STATE

```
#define SC_DET_STATE( ) (HAL_GPIO_ReadPin(SC_det_GPIO_Port, SC_det_Pin))
```

#### 6.23.2.5 WRN\_STATE

```
#define WRN_STATE(
    port,
    pin ) (HAL_GPIO_ReadPin(port, pin))
```

### 6.23.3 Enumeration Type Documentation

#### 6.23.3.1 LEDMode

enum [LEDMode](#)

##### Enumerator

<a href="#">LED_MODE_BLINK_FAST</a>	Fast blink mode
<a href="#">LED_MODE_BLINK_SLOW</a>	Slow blink mode
<a href="#">LED_MODE_ON</a>	<a href="#">LED</a> on mode
<a href="#">LED_MODE_OFF</a>	<a href="#">LED</a> off mode

### 6.23.4 Function Documentation

#### 6.23.4.1 LED\_handler()

```
void LED_handler (
    LED * led,
    uint32_t ms_counter )
```

[LED](#) handler function.

This function handles the [LED](#) blinking modes based on the [LED](#) mode and current millisecond counter.

##### Parameters

<i>led</i>	Pointer to the <a href="#">LED</a> structure.
<i>ms_counter</i>	Millisecond counter for timing.

This function handles the [LED](#) blinking modes based on the [LED](#) mode and current millisecond counter.

##### Parameters

<i>led</i>	Pointer to the <a href="#">LED</a> structure.
<i>ms_counter</i>	Current millisecond counter.

Here is the caller graph for this function:



## 6.23.5 Variable Documentation

### 6.23.5.1 led\_error

```
LED led_error [extern]
```

### 6.23.5.2 led\_left

```
LED led_left [extern]
```

### 6.23.5.3 led\_right

```
LED led_right [extern]
```

## 6.24 PCB\_IO.h

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

```

00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020
00021 #ifndef PCB_IO_H
00022 #define PCB_IO_H
00023
00024 #include "stm32f7xx_hal.h"
00025
00026 // Read SC_det and DIR GPIOs
00027 #define SC_DET_STATE()
00028 #define DIR_STATE()
00029
00030 // Read WRN GPIOs
00031 #define WRN_STATE(port, pin)
00032
00033 // Control ENABLE GPIOs
00034 #define ENABLE(port, pin)
00035 #define DISABLE(port, pin)
00036
00037 // Define LED modes
00038 typedef enum {
00039     LED_MODE_BLINK_FAST,
00040     LED_MODE_BLINK_SLOW,
00041     LED_MODE_ON,
00042     LED_MODE_OFF
00043 } LEDMode;
00044
00048 typedef struct {
00049     GPIO_TypeDef *port;
00050     uint16_t pin;
00051     LEDMode mode;
  
```

```
(HAL_GPIO_ReadPin(SC_det_GPIO_Port, SC_det_Pin))
```

```
(HAL_GPIO_ReadPin(DIR_GPIO_Port, DIR_Pin))
```

```
(HAL_GPIO_ReadPin(port, pin))
```

```
do { HAL_GPIO_WritePin(port, pin, GPIO_PIN_SET); } while(0)
do { HAL_GPIO_WritePin(port, pin, GPIO_PIN_RESET); } while(0)
```

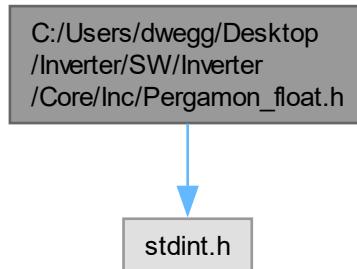
```

00052 } LED;
00053
00054 // Declare LED variables as extern
00055 extern LED led_left;
00056 extern LED led_right;
00057 extern LED led_error;
00058
00059 // Function prototypes
00068 void LED_handler(LED *led, uint32_t ms_counter);
00069
00070 #endif /* PCB_IO_H */

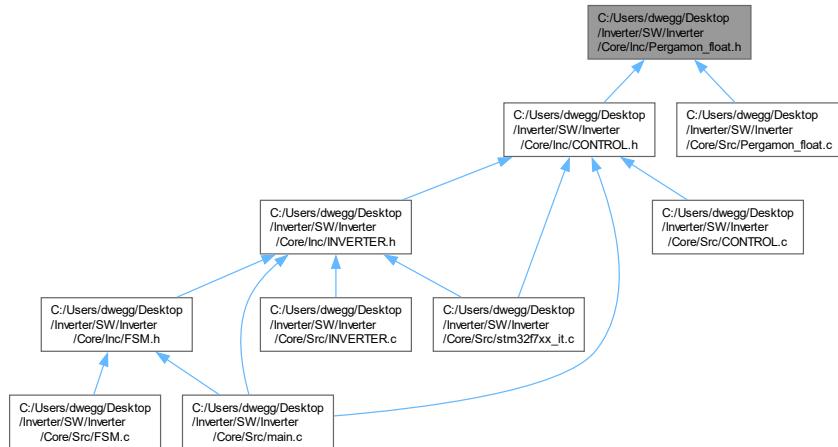
```

## 6.25 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon\_float.h File Reference

#include "stdint.h"  
 Include dependency graph for Pergamon\_float.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct [pi\\_aw\\_struct](#)  
*PI Controller with internal saturation, anti-windup, and feedforward.*
- struct [pi\\_struct](#)  
*PI Controller with external saturation and feedforward.*
- struct [clarke3F\\_struct](#)  
*Clarke transformation for three-phase systems.*
- struct [iclare3F\\_struct](#)  
*Inverse Clarke transformation for three-phase systems.*
- struct [rot\\_struct](#)  
*Rotates the DQ axis in the opposite direction (clockwise).*
- struct [irot\\_struct](#)  
*Inverse rotation (counterclockwise).*
- struct [angle\\_struct](#)  
*Generates an angle based on a fixed frequency.*
- struct [svpwm\\_struct](#)  
*Space Vector Pulse Width Modulation (SVPWM) implementation.*
- struct [rampa\\_struct](#)  
*Single-ramp generator.*
- struct [rampa\\_dual\\_struct](#)  
*Dual-ramp generator.*
- struct [datalog\\_struct](#)
- struct [avg\\_struct\\_10](#)  
*Moving average filter for 10 samples.*
- struct [RMS\\_struct](#)  
*Root Mean Square (RMS) calculation.*
- struct [filtreLP\\_struct](#)  
*First-order low-pass filter.*
- struct [step\\_struct](#)  
*Step function generator.*

## Macros

- #define [SQ2](#) 1.4142135624F
- #define [ISQ2](#) 0.7071067812F
- #define [SQ3](#) 1.7320508076F
- #define [ISQ3](#) 0.5773502692F
- #define [PI](#) 3.1415926536F
- #define [IPI](#) 0.3183098862F
- #define [PI2](#) 6.2831853072F
- #define [IPI2](#) 0.1591549431F
- #define [INV\\_DEG](#) 0.0027777778F
- #define [INV3](#) 0.3333333333F
- #define [DIV2](#) 0.5F
- #define [N\\_DATALOG](#) 256  
*Data logger for logging variables.*

## Functions

- void `pi_aw_calc` (volatile `pi_aw_struct` \*v) `__attribute__((section(".ccmram")))`  
*Initializes the PI controller with anti-windup.*
- void `pi_init` (volatile `pi_struct` \*v) `__attribute__((section(".ccmram")))`  
*Initializes the PI controller.*
- void `pi_calc` (volatile `pi_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the output of the PI controller.*
- void `pi_extsat_calc` (volatile `pi_struct` \*v)  
*Calculates the output of the PI controller with external saturation.*
- void `clarke3F_calc` (volatile `clarke3F_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the Clarke transformation.*
- void `iclarke3F_calc` (volatile `iclarke3F_struct` \*v)  
*Calculates the inverse Clarke transformation.*
- void `rot_calc` (volatile `rot_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the rotation in the opposite direction.*
- void `irot_calc` (volatile `irot_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the inverse rotation.*
- void `angle_calc` (volatile `angle_struct` \*p) `__attribute__((section(".ccmram")))`  
*Calculates the angle generation.*
- void `svpwm_calc` (volatile `svpwm_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the SVPWM outputs.*
- void `rampa_calc` (volatile `rampa_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the output of the single ramp generator.*
- void `rampa_dual_calc` (volatile `rampa_dual_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the output of the dual ramp generator.*
- void `datalog_calc` (volatile `datalog_struct` \*dl)  
*Calculates the data log.*
- void `avg_calc_10_samples` (volatile `avg_struct_10` \*v)  
*Calculates the moving average for 10 samples.*
- void `RMS_calc` (volatile `RMS_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the RMS.*
- void `filtreLP_init` (volatile `filtreLP_struct` \*v)  
*Initializes the first-order low-pass filter.*
- void `filtreLP_calc` (volatile `filtreLP_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the output of the first-order low-pass filter.*
- void `step_calc` (volatile `step_struct` \*v) `__attribute__((section(".ccmram")))`  
*Calculates the output of the step function generator.*

## 6.26 Pergamon\_float.h

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

```

00001 /*
00002  * @file pergammon_float.h
00003  * @brief Library consolidating all the hardware-independent libraries of CITCEA.
00004  * This library is based on Alexandria created by Gabriel Gross and Quim Lopez Mestre in September
00005  * 2006.
00006  * Created by Gabriel Gross, Daniel Heredero, and Tomas Lledo in December 2015.
00007  * Translated to floats by Lucas Bouzon in March 2020.
00008  */
00009 */
0010
0011
0012

```



```

00013
00014
00015
00016
00017
00018
00019 */
00020
00021 #ifndef __PERGAMON_FLOAT_H__
00022 #define __PERGAMON_FLOAT_H__
00023
00024
00025 #include "stdint.h"
00026
00027 #define SQ2      1.4142135624F
00028 #define ISQ2     0.7071067812F
00029 #define SQ3      1.7320508076F
00030 #define ISQ3     0.5773502692F
00031 #define PI       3.1415926536F
00032 #define IPI      0.3183098862F
00033 #define PI2      6.2831853072F
00034 #define IPI2     0.1591549431F
00035 #define INV_DEG  0.0027777778F
00036 #define INV3      0.3333333333F
00037 #define DIV2      0.5F
00038
00039
00040
00041
00042
00043
00044
00045
00046
00047
00048
00049
00050
00051 typedef struct
00052 {
00053     uint16_t    enable;
00054     float        Ts;
00055     float        Kp;
00056     float        Ki;
00057     float        Kaw;
00058     float        e[2];
00059     float        pi_consig;
00060     float        pi_fdb;
00061     float        pi_out_max;
00062     float        pi_out_min;
00063     float        pi_out_presat;
00064     float        pi_out_postsat;
00065     float        pi_out;
00066     float        pi_int[2];
00067     float        pi_ffw[2];
00068     void(*calc)();
00069 } pi_aw_struct;
00070
00071 void pi_aw_calc(volatile pi_aw_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00072
00073
00074
00075
00076
00077
00078
00079
00080 typedef struct
00081 {
00082     uint16_t    enable;
00083     float        Ts;
00084     float        Kp;
00085     float        Ki;
00086     float        K0;
00087     float        K1;
00088     float        e[2];
00089     float        pi_consig;
00090     float        pi_fdb;
00091     float        pi_out_max;
00092     float        pi_out_min;
00093     float        pi_out;
00094     float        pi_ffw[2];
00095     void(*init)();
00096     void(*calc)();
00097 } pi_struct;
00098
00099
00100
00101
00102
00103 void pi_init(volatile pi_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00104
00105
00106
00107
00108
00109 void pi_calc(volatile pi_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00110
00111
00112
00113
00114
00115 void pi_extsat_calc(volatile pi_struct *v);
00116
00117
00118
00119
00120
00121
00122
00123
00124
00125
00126
00127 typedef struct
00128 {
00129     float        a;
00130     float        b;
00131     float        D;
00132     float        Q;
00133     void(*calc)();
00134 } clarke3F_struct;
00135
00136
00137
00138
00139
00140 void clarke3F_calc(volatile clarke3F_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00141
00142
00143
00144
00145 typedef struct
00146 {
00147     float        D;
00148     float        Q;

```

```

00149     float      a;
00150     float      b;
00151     void(*calc)();
00152 } iclarke3F_struct;
00153
00158 void iclarke3F_calc(volatile iclarke3F_struct *v);
00159
00163 typedef struct
00164 {
00165     float      D;
00166     float      Q;
00167     float      sinFi;
00168     float      cosFi;
00169     float      d;
00170     float      q;
00171     void(*calc)();
00172 } rot_struct;
00173
00178 void rot_calc(volatile rot_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00179
00183 typedef struct
00184 {
00185     float      d;
00186     float      q;
00187     float      sinFi;
00188     float      cosFi;
00189     float      D;
00190     float      Q;
00191     void(*calc)();
00192 } irot_struct;
00193
00198 void irot_calc(volatile irot_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00199
00200 typedef struct {
00201     float      freq;
00202     float      Ts;
00203     float      angle;
00204     void(*calc)();
00205 } angle_struct;
00206
00221 void angle_calc(volatile angle_struct *p) __attribute__( ( section ( ".ccmram" ) ) );
00222
00226 typedef struct
00227 {
00228     float      valfa;
00229     float      vbeta;
00230     float      Ta;
00231     float      Tb;
00232     float      Tc;
00233     void(*calc)();
00234 } svpwm_struct;
00235
00240 void svpwm_calc(volatile svpwm_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00241
00251 typedef struct
00252 {
00253     float      in;
00254     float      out;
00255     float      Incr;
00256     uint8_t    enable;
00257     void(*calc)();
00258 } rampa_struct;
00259
00263 typedef struct
00264 {
00265     float      in;
00266     float      out;
00267     float      Incr;
00268     float      Decr;
00269     uint8_t    enable;
00270     void(*calc)();
00271 } rampa_dual_struct;
00272
00277 void rampa_calc(volatile rampa_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00278
00283 void rampa_dual_calc(volatile rampa_dual_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00284
00289 #define N_DATALOG 256
00291 typedef struct
00292 {
00293     uint16_t   i;
00294     uint16_t   j;
00295     uint16_t   estat;
00296     uint16_t   prescaler;
00297     float     *var;
00298     void(*calc)();
00299     float     log[N_DATALOG];

```

```

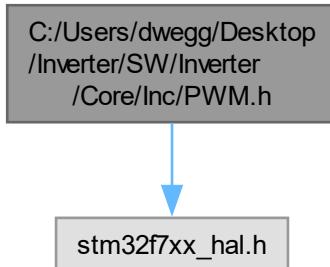
00300 } datalog_struct;
00301
00306 void datalog_calc(volatile datalog_struct *dl);
00307
00311 typedef struct
00312 {
00313     float      out;
00314     float      in[10];
00315 } avg_struct_10;
00316
00321 void avg_calc_10_samples(volatile avg_struct_10 *v);
00322
00326 typedef struct {
00327     float      T_exec;
00328     float      Measure;
00329     float      Sq_Sum;
00330     float      Out_RMS;
00331     float      Freq;
00332     float      Angle;
00333     float      Angle_ant;
00334 } RMS_struct;
00335
00340 void RMS_calc(volatile RMS_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00341
00345 typedef struct
00346 {
00347     float      in;
00348     float      out;
00349     float      alfa;
00350     float      Ts;
00351     float      fc;
00352     uint16_t   enable;
00353     void(*init)();
00354     void(*calc)();
00355 } filtreLP_struct;
00356
00361 void filtreLP_init(volatile filtreLP_struct *v);
00362
00367 void filtreLP_calc(volatile filtreLP_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00368
00379 typedef struct
00380 {
00381     float      fs;
00382     float      In;
00383     float      Out;
00384     float      Step;
00385     float      t_step;
00386     uint32_t   Pulses;
00387     uint32_t   Counter;
00388     uint16_t   enable;
00389     void(*calc)();
00390 } step_struct;
00391
00396 void step_calc(volatile step_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00397
00400 #endif // __PERGAMON_FLOAT_H__

```

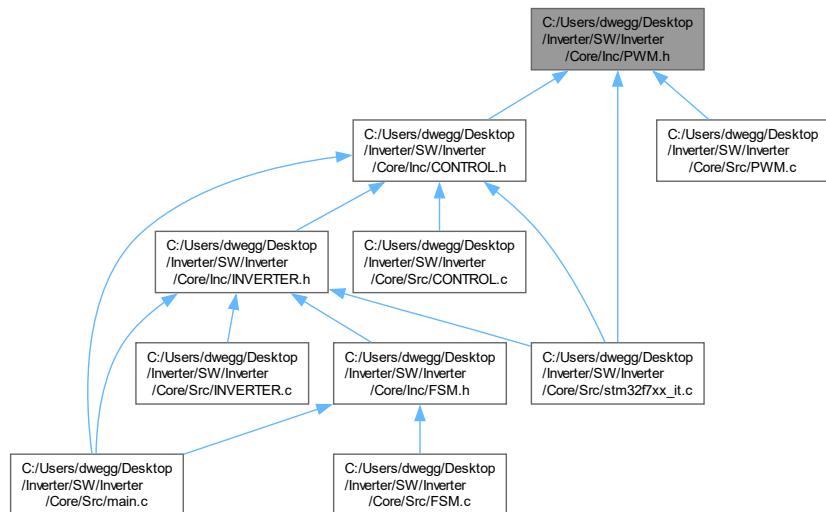
## 6.27 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PWM.h File Reference

Header file for controlling PWM output.

```
#include "stm32f7xx_hal.h"
Include dependency graph for PWM.h:
```



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct [Duties](#)  
*Structure to hold PWM configuration parameters.*
- struct [Compare3F\\_struct](#)

## Functions

- void [enable\\_PWM](#) (TIM\_HandleTypeDef \*htim)  
*Enable PWM output.*
- void [disable\\_PWM](#) (TIM\_HandleTypeDef \*htim)

*Disable PWM output.*

- void **Compare3F\_calc** (TIM\_HandleTypeDef \*htim, Compare3F\_struct \*v)
- void **set\_PWM** (TIM\_HandleTypeDef \*htim, Duties duties)

*Set PWM duty cycles.*

## Variables

- Compare3F\_struct compares\_LEFT
- Compare3F\_struct compares\_RIGHT

### 6.27.1 Detailed Description

Header file for controlling PWM output.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg in GitHub). All rights reserved.

This software is licensed under the Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. For more information, see: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

### 6.27.2 Function Documentation

#### 6.27.2.1 Compare3F\_calc()

```
void Compare3F_calc (
    TIM_HandleTypeDef * htim,
    Compare3F_struct * v )
```

Here is the caller graph for this function:



#### 6.27.2.2 disable\_PWM()

```
void disable_PWM (
    TIM_HandleTypeDef * htim )
```

Disable PWM output.

This function disables PWM output for the specified timer.

**Parameters**

<i>htim</i>	Pointer to the TIM_HandleTypeDef structure.
-------------	---

**6.27.2.3 enable\_PWM()**

```
void enable_PWM (
    TIM_HandleTypeDef * htim )
```

Enable PWM output.

This function enables PWM output for the specified timer.

**Parameters**

<i>htim</i>	Pointer to the TIM_HandleTypeDef structure.
-------------	---

Here is the caller graph for this function:

**6.27.2.4 set\_PWM()**

```
void set_PWM (
    TIM_HandleTypeDef * htim,
    Duties duties )
```

Set PWM duty cycles.

This function sets the duty cycles for the PWM channels.

**Parameters**

<i>htim</i>	Pointer to the TIM_HandleTypeDef structure.
<i>duties</i>	Duties structure containing duty cycle values.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.27.3 Variable Documentation

#### 6.27.3.1 compares\_LEFT

`Compare3F_struct compares_LEFT [extern]`

Compares for the left side

#### 6.27.3.2 compares\_RIGHT

`Compare3F_struct compares_RIGHT [extern]`

Compares for the right side

## 6.28 PWM.h

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

```

00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020 #ifndef PWM_H
00021 #define PWM_H
00022
00023 #include "stm32f7xx_hal.h"
00024
00028 typedef struct {
00029     float Da;
00030     float Db;
00031     float Dc;
00032 } Duties;
  
```

```

00033
00041 void enable_PWM(TIM_HandleTypeDef *htim);
00042
00050 void disable_PWM(TIM_HandleTypeDef *htim);
00051
00052 typedef struct
00053 {
00054     float      alfaA;      // Alfa d'entrada, branca A, en PU
00055     float      alfaB;      // Alfa d'entrada, branca B, en PU
00056     float      alfaC;      // Alfa d'entrada, branca C, en PU
00057     uint32_t   compA;      // Valor de sortida a aplicar al CompareA
00058     uint32_t   compB;      // Valor de sortida a aplicar al CompareB
00059     uint32_t   compC;      // Valor de sortida a aplicar al CompareC
00060     void(*calc)();        // Punter a la funcio per calcular 3 Compares
00061 } Compare3F_struct;
00062
00063 extern Compare3F_struct compares_LEFT;
00064 extern Compare3F_struct compares_RIGHT;
00065 void Compare3F_calc(TIM_HandleTypeDef *htim, Compare3F_struct *v);
00066
00076 void set_PWM(TIM_HandleTypeDef *htim, Duties duties);
00077
00078 #endif /* PWM_H */

```

## 6.29 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/stm32f7xx\_hal\_conf.h File Reference

```

#include "stm32f7xx_hal_rcc.h"
#include "stm32f7xx_hal_exti.h"
#include "stm32f7xx_hal_gpio.h"
#include "stm32f7xx_hal_dma.h"
#include "stm32f7xx_hal_cortex.h"
#include "stm32f7xx_hal_adc.h"
#include "stm32f7xx_hal_can.h"
#include "stm32f7xx_hal_dac.h"
#include "stm32f7xx_hal_flash.h"
#include "stm32f7xx_hal_i2c.h"
#include "stm32f7xx_hal_pwr.h"
#include "stm32f7xx_hal_tim.h"

```

Include dependency graph for stm32f7xx\_hal\_conf.h:



### Macros

- **#define HAL\_MODULE\_ENABLED**  
*This is the list of modules to be used in the HAL driver.*
- **#define HAL\_ADC\_MODULE\_ENABLED**
- **#define HAL\_CAN\_MODULE\_ENABLED**
- **#define HAL\_DAC\_MODULE\_ENABLED**
- **#define HAL\_TIM\_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\_I2C\_MODULE\_ENABLED
- #define HAL\_CORTEX\_MODULE\_ENABLED
- #define HSE\_VALUE ((uint32\_t)20000000U)
 

*Adjust the value of External High Speed oscillator (HSE) used in your application. This value is used by the RCC HAL module to compute the system frequency (when HSE is used as system clock source, directly or through the PLL).*
- #define HSE\_STARTUP\_TIMEOUT ((uint32\_t)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 ((uint32\_t)32000U)
 

*Internal Low Speed oscillator (LSI) value.*
- #define LSE\_VALUE ((uint32\_t)32768U)
 

*External Low Speed oscillator (LSE) value.*
- #define LSE\_STARTUP\_TIMEOUT ((uint32\_t)5000U)
- #define EXTERNAL\_CLOCK\_VALUE ((uint32\_t)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 ((uint32\_t)15U)
- #define USERTOS 0U
- #define PREFETCH\_ENABLE 0U
- #define ART\_ACCELERATOR\_ENABLE 0U /\* To enable instruction cache and prefetch \*/
- #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\_I2S\_REGISTER\_CALLBACKS 0U /\* I2S register callback disabled \*/
- #define USE\_HAL\_IRDA\_REGISTER\_CALLBACKS 0U /\* IRDA register callback disabled \*/
- #define USE\_HAL\_JPEG\_REGISTER\_CALLBACKS 0U /\* JPEG 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\_MDIOS\_REGISTER\_CALLBACKS 0U /\* MDIOS register callback disabled \*/
- #define USE\_HAL\_MMCHS\_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\_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\_RXBUFN ((uint32\_t)4U) /\* 4 Rx buffers of size `ETH_RX_BUF_SIZE` \*/
- #define ETH\_TXBUFN ((uint32\_t)4U) /\* 4 Tx buffers of size `ETH_TX_BUF_SIZE` \*/
- #define DP83848\_PHY\_ADDRESS
- #define PHY\_RESET\_DELAY ((uint32\_t)0x000000FFU)
- #define PHY\_CONFIG\_DELAY ((uint32\_t)0x00000FFFU)
- #define PHY\_READ\_TO ((uint32\_t)0x0000FFFFU)
- #define PHY\_WRITE\_TO ((uint32\_t)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.*

## 6.29.1 Macro Definition Documentation

### 6.29.1.1 ART\_ACCELERATOR\_ENABLE

```
#define ART_ACCELERATOR_ENABLE 0U /* To enable instruction cache and prefetch */
```

### 6.29.1.2 assert\_param

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

Include module's header file.

### 6.29.1.3 DP83848\_PHY\_ADDRESS

```
#define DP83848_PHY_ADDRESS
```

### 6.29.1.4 ETH\_RX\_BUF\_SIZE

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

### 6.29.1.5 ETH\_RXBUFN

```
#define ETH_RXBUFN ((uint32_t)4U) /* 4 Rx buffers of size ETH_RX_BUF_SIZE */
```

### 6.29.1.6 ETH\_TX\_BUF\_SIZE

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

### 6.29.1.7 ETH\_TXBUFN

```
#define ETH_TXBUFN ((uint32_t)4U) /* 4 Tx buffers of size ETH_TX_BUF_SIZE */
```

### 6.29.1.8 EXTERNAL\_CLOCK\_VALUE

```
#define EXTERNAL_CLOCK_VALUE ((uint32_t)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 Internal oscillator in Hz

### 6.29.1.9 HAL\_ADC\_MODULE\_ENABLED

```
#define HAL_ADC_MODULE_ENABLED
```

### 6.29.1.10 HAL\_CAN\_MODULE\_ENABLED

```
#define HAL_CAN_MODULE_ENABLED
```

### 6.29.1.11 HAL\_CORTEX\_MODULE\_ENABLED

```
#define HAL_CORTEX_MODULE_ENABLED
```

### 6.29.1.12 HAL\_DAC\_MODULE\_ENABLED

```
#define HAL_DAC_MODULE_ENABLED
```

### 6.29.1.13 HAL\_DMA\_MODULE\_ENABLED

```
#define HAL_DMA_MODULE_ENABLED
```

### 6.29.1.14 HAL\_EXTI\_MODULE\_ENABLED

```
#define HAL_EXTI_MODULE_ENABLED
```

### 6.29.1.15 HAL\_FLASH\_MODULE\_ENABLED

```
#define HAL_FLASH_MODULE_ENABLED
```

### 6.29.1.16 HAL\_GPIO\_MODULE\_ENABLED

```
#define HAL_GPIO_MODULE_ENABLED
```

### 6.29.1.17 HAL\_I2C\_MODULE\_ENABLED

```
#define HAL_I2C_MODULE_ENABLED
```

### 6.29.1.18 HAL\_MODULE\_ENABLED

```
#define HAL_MODULE_ENABLED
```

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

### 6.29.1.19 HAL\_PWR\_MODULE\_ENABLED

```
#define HAL_PWR_MODULE_ENABLED
```

### 6.29.1.20 HAL\_RCC\_MODULE\_ENABLED

```
#define HAL_RCC_MODULE_ENABLED
```

### 6.29.1.21 HAL\_TIM\_MODULE\_ENABLED

```
#define HAL_TIM_MODULE_ENABLED
```

### 6.29.1.22 HSE\_STARTUP\_TIMEOUT

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

Time out for HSE start up, in ms

### 6.29.1.23 HSE\_VALUE

```
#define HSE_VALUE ((uint32_t)20000000U)
```

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

### 6.29.1.24 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

### 6.29.1.25 LSE\_STARTUP\_TIMEOUT

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

Time out for LSE start up, in ms

### 6.29.1.26 LSE\_VALUE

```
#define LSE_VALUE ((uint32_t)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

**6.29.1.27 LSI\_VALUE**

```
#define LSI_VALUE ((uint32_t)32000U)
```

Internal Low Speed oscillator (LSI) value.

LSI Typical Value in Hz

**6.29.1.28 MAC\_ADDR0**

```
#define MAC_ADDR0 2U
```

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

**6.29.1.29 MAC\_ADDR1**

```
#define MAC_ADDR1 0U
```

**6.29.1.30 MAC\_ADDR2**

```
#define MAC_ADDR2 0U
```

**6.29.1.31 MAC\_ADDR3**

```
#define MAC_ADDR3 0U
```

**6.29.1.32 MAC\_ADDR4**

```
#define MAC_ADDR4 0U
```

**6.29.1.33 MAC\_ADDR5**

```
#define MAC_ADDR5 0U
```

**6.29.1.34 PHY\_AUTONEGO\_COMPLETE**

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

Auto-Negotiation process completed

### 6.29.1.35 PHY\_AUTONEGOTIATION

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

Enable auto-negotiation function

### 6.29.1.36 PHY\_BCR

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

Transceiver Basic Control Register

### 6.29.1.37 PHY\_BSR

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

Transceiver Basic Status Register

### 6.29.1.38 PHY\_CONFIG\_DELAY

```
#define PHY_CONFIG_DELAY ((uint32_t)0x00000FFFU)
```

### 6.29.1.39 PHY\_DUPLEX\_STATUS

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

PHY Duplex mask

### 6.29.1.40 PHY\_FULLDUPLEX\_100M

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

Set the full-duplex mode at 100 Mb/s

### 6.29.1.41 PHY\_FULLDUPLEX\_10M

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

Set the full-duplex mode at 10 Mb/s

**6.29.1.42 PHY\_HALFDUPLEX\_100M**

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

Set the half-duplex mode at 100 Mb/s

**6.29.1.43 PHY\_HALFDUPLEX\_10M**

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

Set the half-duplex mode at 10 Mb/s

**6.29.1.44 PHY\_ISOLATE**

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

Isolate PHY from MII

**6.29.1.45 PHY\_JABBER\_DETECTION**

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

Jabber condition detected

**6.29.1.46 PHY\_LINKED\_STATUS**

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

Valid link established

**6.29.1.47 PHY\_LOOPBACK**

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

Select loop-back mode

**6.29.1.48 PHY\_POWERDOWN**

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

Select the power down mode

**6.29.1.49 PHY\_READ\_TO**

```
#define PHY_READ_TO ((uint32_t)0x0000FFFFU)
```

**6.29.1.50 PHY\_RESET**

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

PHY Reset

**6.29.1.51 PHY\_RESET\_DELAY**

```
#define PHY_RESET_DELAY ((uint32_t)0x000000FFU)
```

**6.29.1.52 PHY\_RESTART\_AUTONEGOTIATION**

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

Restart auto-negotiation function

**6.29.1.53 PHY\_SPEED\_STATUS**

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

PHY Speed mask

**6.29.1.54 PHY\_SR**

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

PHY status register Offset

**6.29.1.55 PHY\_WRITE\_TO**

```
#define PHY_WRITE_TO ((uint32_t)0x0000FFFFU)
```

**6.29.1.56 PREFETCH\_ENABLE**

```
#define PREFETCH_ENABLE 0U
```

**6.29.1.57 TICK\_INT\_PRIORITY**

```
#define TICK_INT_PRIORITY ((uint32_t)15U)
```

tick interrupt priority

**6.29.1.58 USE\_HAL\_ADC\_REGISTER\_CALLBACKS**

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

**6.29.1.59 USE\_HAL\_CAN\_REGISTER\_CALLBACKS**

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

**6.29.1.60 USE\_HAL\_CEC\_REGISTER\_CALLBACKS**

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

**6.29.1.61 USE\_HAL\_CRYP\_REGISTER\_CALLBACKS**

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

**6.29.1.62 USE\_HAL\_DAC\_REGISTER\_CALLBACKS**

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

**6.29.1.63 USE\_HAL\_DCMI\_REGISTER\_CALLBACKS**

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

**6.29.1.64 USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS**

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

**6.29.1.65 USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS**

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

**6.29.1.66 USE\_HAL\_DSI\_REGISTER\_CALLBACKS**

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

### 6.29.1.67 USE\_HAL\_ETH\_REGISTER\_CALLBACKS

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

### 6.29.1.68 USE\_HAL\_HASH\_REGISTER\_CALLBACKS

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

### 6.29.1.69 USE\_HAL\_HCD\_REGISTER\_CALLBACKS

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

### 6.29.1.70 USE\_HAL\_I2C\_REGISTER\_CALLBACKS

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

### 6.29.1.71 USE\_HAL\_I2S\_REGISTER\_CALLBACKS

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

### 6.29.1.72 USE\_HAL\_IRDA\_REGISTER\_CALLBACKS

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

### 6.29.1.73 USE\_HAL\_JPEG\_REGISTER\_CALLBACKS

```
#define USE_HAL_JPEG_REGISTER_CALLBACKS 0U /* JPEG register callback disabled */
```

### 6.29.1.74 USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS

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

### 6.29.1.75 USE\_HAL\_LTDC\_REGISTER\_CALLBACKS

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

### 6.29.1.76 USE\_HAL\_MDIOS\_REGISTER\_CALLBACKS

```
#define USE_HAL_MDIOS_REGISTER_CALLBACKS 0U /* MDIOS register callback disabled */
```

### 6.29.1.77 USE\_HAL\_MMC\_REGISTER\_CALLBACKS

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

### 6.29.1.78 USE\_HAL\_NAND\_REGISTER\_CALLBACKS

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

### 6.29.1.79 USE\_HAL\_NOR\_REGISTER\_CALLBACKS

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

### 6.29.1.80 USE\_HAL\_PCD\_REGISTER\_CALLBACKS

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

### 6.29.1.81 USE\_HAL\_QSPI\_REGISTER\_CALLBACKS

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

### 6.29.1.82 USE\_HAL\_RNG\_REGISTER\_CALLBACKS

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

### 6.29.1.83 USE\_HAL\_RTC\_REGISTER\_CALLBACKS

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

### 6.29.1.84 USE\_HAL\_SAI\_REGISTER\_CALLBACKS

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

### 6.29.1.85 USE\_HAL\_SD\_REGISTER\_CALLBACKS

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

### 6.29.1.86 USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### **6.29.1.96 USE\_RTOS**

```
#define USE_RTOS 0U
```

### 6.29.1.97 USE\_SPI\_CRC

```
#define USE_SPI_CRC 0U
```

### 6.29.1.98 VDD\_VALUE

```
#define VDD_VALUE 3300U
```

This is the HAL system configuration section.

Value of VDD in mv

## 6.30 stm32f7xx\_hal\_conf.h

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

```
00001 /* USER CODE BEGIN Header */
00021 /* USER CODE END Header */
00022
00023 /* Define to prevent recursive inclusion -----*/
00024 #ifndef __STM32F7XX_HAL_CONF_H
00025 #define __STM32F7XX_HAL_CONF_H
00026
00027 #ifdef __cplusplus
00028 extern "C" {
00029 #endif
00030
00031 /* Exported types -----*/
00032 /* Exported constants -----*/
00033
00034 /* ##### Module Selection ##### */
00038 #define HAL_MODULE_ENABLED
00039
00040 /* #define HAL_CRYPT_MODULE_ENABLED */
00041 #define HAL_ADC_MODULE_ENABLED
00042 #define HAL_CAN_MODULE_ENABLED
00043 /* #define HAL_CEC_MODULE_ENABLED */
00044 /* #define HAL_CRC_MODULE_ENABLED */
00045 #define HAL_DAC_MODULE_ENABLED
00046 /* #define HAL_DCMI_MODULE_ENABLED */
00047 /* #define HAL_DMA2D_MODULE_ENABLED */
00048 /* #define HAL_ETH_MODULE_ENABLED */
00049 /* #define HAL_ETH_LEGACY_MODULE_ENABLED */
00050 /* #define HAL_NAND_MODULE_ENABLED */
00051 /* #define HAL_NOR_MODULE_ENABLED */
00052 /* #define HAL_SRAM_MODULE_ENABLED */
00053 /* #define HAL_SDRAM_MODULE_ENABLED */
00054 /* #define HAL_HASH_MODULE_ENABLED */
00055 /* #define HAL_I2S_MODULE_ENABLED */
00056 /* #define HAL_IWDG_MODULE_ENABLED */
00057 /* #define HAL_LPTIM_MODULE_ENABLED */
00058 /* #define HAL_LTDC_MODULE_ENABLED */
00059 /* #define HAL_QSPI_MODULE_ENABLED */
00060 /* #define HAL_RNG_MODULE_ENABLED */
00061 /* #define HAL_RTC_MODULE_ENABLED */
00062 /* #define HAL_SAI_MODULE_ENABLED */
00063 /* #define HAL_SD_MODULE_ENABLED */
00064 /* #define HAL_MMC_MODULE_ENABLED */
00065 /* #define HAL_SPDIFRX_MODULE_ENABLED */
00066 /* #define HAL_SPI_MODULE_ENABLED */
00067 #define HAL_TIM_MODULE_ENABLED
00068 /* #define HAL_UART_MODULE_ENABLED */
00069 /* #define HAL_USART_MODULE_ENABLED */
00070 /* #define HAL_IRDA_MODULE_ENABLED */
00071 /* #define HAL_SMARTCARD_MODULE_ENABLED */
00072 /* #define HAL_WWDG_MODULE_ENABLED */
00073 /* #define HAL_PCD_MODULE_ENABLED */
00074 /* #define HAL_HCD_MODULE_ENABLED */
00075 /* #define HAL_DFSDM_MODULE_ENABLED */
00076 /* #define HAL_DSI_MODULE_ENABLED */
00077 /* #define HAL_JPEG_MODULE_ENABLED */
00078 /* #define HAL_MDIOS_MODULE_ENABLED */
00079 /* #define HAL_SMBUS_MODULE_ENABLED */
00080 /* #define HAL_EXTI_MODULE_ENABLED */
```

```

00081 #define HAL_GPIO_MODULE_ENABLED
00082 #define HAL_EXTI_MODULE_ENABLED
00083 #define HAL_DMA_MODULE_ENABLED
00084 #define HAL_RCC_MODULE_ENABLED
00085 #define HAL_FLASH_MODULE_ENABLED
00086 #define HAL_PWR_MODULE_ENABLED
00087 #define HAL_I2C_MODULE_ENABLED
00088 #define HAL_CORTEX_MODULE_ENABLED
00089
00090 /* ##### HSE/HSI Values adaptation ##### */
00096 #if !defined (HSE_VALUE)
00097 #define HSE_VALUE ((uint32_t)20000000U)
00098 #endif /* HSE_VALUE */
00099
00100 #if !defined (HSE_STARTUP_TIMEOUT)
00101 #define HSE_STARTUP_TIMEOUT ((uint32_t)100U)
00102 #endif /* HSE_STARTUP_TIMEOUT */
00103
00109 #if !defined (HSI_VALUE)
00110 #define HSI_VALUE ((uint32_t)16000000U)
00111 #endif /* HSI_VALUE */
00112
00116 #if !defined (LSI_VALUE)
00117 #define LSI_VALUE ((uint32_t)32000U)
00118 #endif /* LSI_VALUE */
00124 #if !defined (LSE_VALUE)
00125 #define LSE_VALUE ((uint32_t)32768U)
00126 #endif /* LSE_VALUE */
00127
00128 #if !defined (LSE_STARTUP_TIMEOUT)
00129 #define LSE_STARTUP_TIMEOUT ((uint32_t)5000U)
00130 #endif /* LSE_STARTUP_TIMEOUT */
00131
00137 #if !defined (EXTERNAL_CLOCK_VALUE)
00138 #define EXTERNAL_CLOCK_VALUE ((uint32_t)12288000U)
00139 #endif /* EXTERNAL_CLOCK_VALUE */
00140
00141 /* Tip: To avoid modifying this file each time you need to use different HSE,
00142 === you can define the HSE value in your toolchain compiler preprocessor. */
00143
00144 /* ##### System Configuration ##### */
00148 #define VDD_VALUE 3300U
00149 #define TICK_INT_PRIORITY ((uint32_t)15U)
00150 #define USERTOS 0U
00151 #define PREFETCH_ENABLE 0U
00152 #define ART_ACCELERATOR_ENABLE 0U /* To enable instruction cache and prefetch */
00153
00154 #define USE_HAL_ADC_REGISTER_CALLBACKS 0U /* ADC register callback disabled */
00155 #define USE_HAL_CAN_REGISTER_CALLBACKS 0U /* CAN register callback disabled */
00156 #define USE_HAL_CEC_REGISTER_CALLBACKS 0U /* CEC register callback disabled */
00157 #define USE_HAL_CRYP_REGISTER_CALLBACKS 0U /* CRYP register callback disabled */
00158 #define USE_HAL_DAC_REGISTER_CALLBACKS 0U /* DAC register callback disabled */
00159 #define USE_HAL_DCMI_REGISTER_CALLBACKS 0U /* DCMI register callback disabled */
00160 #define USE_HAL_DFSDM_REGISTER_CALLBACKS 0U /* DFSDM register callback disabled */
00161 #define USE_HAL_DMA2D_REGISTER_CALLBACKS 0U /* DMA2D register callback disabled */
00162 #define USE_HAL_DSI_REGISTER_CALLBACKS 0U /* DSI register callback disabled */
00163 #define USE_HAL_ETH_REGISTER_CALLBACKS 0U /* ETH register callback disabled */
00164 #define USE_HAL_HASH_REGISTER_CALLBACKS 0U /* HASH register callback disabled */
00165 #define USE_HAL_HCD_REGISTER_CALLBACKS 0U /* HCD register callback disabled */
00166 #define USE_HAL_I2C_REGISTER_CALLBACKS 0U /* I2C register callback disabled */
00167 #define USE_HAL_I2S_REGISTER_CALLBACKS 0U /* I2S register callback disabled */
00168 #define USE_HAL_IRDA_REGISTER_CALLBACKS 0U /* IRDA register callback disabled */
00169 #define USE_HAL_JPEG_REGISTER_CALLBACKS 0U /* JPEG register callback disabled */
00170 #define USE_HAL_LPTIM_REGISTER_CALLBACKS 0U /* LPTIM register callback disabled */
00171 #define USE_HAL_LTDC_REGISTER_CALLBACKS 0U /* LTDC register callback disabled */
00172 #define USE_HAL_MDIOS_REGISTER_CALLBACKS 0U /* MDIOS register callback disabled */
00173 #define USE_HAL_MMCHOST_REGISTER_CALLBACKS 0U /* MMC register callback disabled */
00174 #define USE_HAL_NAND_REGISTER_CALLBACKS 0U /* NAND register callback disabled */
00175 #define USE_HAL_NOR_REGISTER_CALLBACKS 0U /* NOR register callback disabled */
00176 #define USE_HAL_PCD_REGISTER_CALLBACKS 0U /* PCD register callback disabled */
00177 #define USE_HAL_QSPI_REGISTER_CALLBACKS 0U /* QSPI register callback disabled */
00178 #define USE_HAL_RNG_REGISTER_CALLBACKS 0U /* RNG register callback disabled */
00179 #define USE_HAL_RTC_REGISTER_CALLBACKS 0U /* RTC register callback disabled */
00180 #define USE_HAL_SAI_REGISTER_CALLBACKS 0U /* SAI register callback disabled */
00181 #define USE_HAL_SD_REGISTER_CALLBACKS 0U /* SD register callback disabled */
00182 #define USE_HAL_SMARTCARD_REGISTER_CALLBACKS 0U /* SMARTCARD register callback disabled */
00183 #define USE_HAL_SDRAM_REGISTER_CALLBACKS 0U /* SDRAM register callback disabled */
00184 #define USE_HAL_SRAM_REGISTER_CALLBACKS 0U /* SRAM register callback disabled */
00185 #define USE_HAL_SPDIFRX_REGISTER_CALLBACKS 0U /* SPDIFRX register callback disabled */
00186 #define USE_HAL_SMBUS_REGISTER_CALLBACKS 0U /* SMBUS register callback disabled */
00187 #define USE_HAL_SPI_REGISTER_CALLBACKS 0U /* SPI register callback disabled */
00188 #define USE_HAL_TIM_REGISTER_CALLBACKS 0U /* TIM register callback disabled */
00189 #define USE_HAL_UART_REGISTER_CALLBACKS 0U /* UART register callback disabled */
00190 #define USE_HAL_USART_REGISTER_CALLBACKS 0U /* USART register callback disabled */
00191 #define USE_HAL_WWDG_REGISTER_CALLBACKS 0U /* WWDG register callback disabled */
00192
00193 /* ##### Assert Selection ##### */

```

```

00198 /* #define USE_FULL_ASSERT      1U */
00199
00200 /* ##### Ethernet peripheral configuration ###### */
00201
00202 /* Section 1 : Ethernet peripheral configuration */
00203
00204 /* MAC ADDRESS: MAC_ADDR0:MAC_ADDR1:MAC_ADDR2:MAC_ADDR3:MAC_ADDR4:MAC_ADDR5 */
00205 #define MAC_ADDR0  2U
00206 #define MAC_ADDR1  0U
00207 #define MAC_ADDR2  0U
00208 #define MAC_ADDR3  0U
00209 #define MAC_ADDR4  0U
00210 #define MAC_ADDR5  0U
00211
00212 /* Definition of the Ethernet driver buffers size and count */
00213 #define ETH_RX_BUF_SIZE          ETH_MAX_PACKET_SIZE /* buffer size for receive */
00214 #define ETH_TX_BUF_SIZE          ETH_MAX_PACKET_SIZE /* buffer size for transmit */
00215 #define ETH_RXBUFN  ((uint32_t)4U)    /* 4 Rx buffers of size ETH_RX_BUF_SIZE */
00216 #define ETH_TXBUFN  ((uint32_t)4U)    /* 4 Tx buffers of size ETH_TX_BUF_SIZE */
00217
00218 /* Section 2: PHY configuration section */
00219
00220 /* DP83848_PHY_ADDRESS Address*/
00221 #define DP83848_PHY_ADDRESS
00222 /* PHY Reset delay these values are based on a 1 ms Systick interrupt*/
00223 #define PHY_RESET_DELAY        ((uint32_t)0x000000FFU)
00224 /* PHY Configuration delay */
00225 #define PHY_CONFIG_DELAY       ((uint32_t)0x000000FFU)
00226
00227 #define PHY_READ_TO           ((uint32_t)0x0000FFFFU)
00228 #define PHY_WRITE_TO          ((uint32_t)0x0000FFFFU)
00229
00230 /* Section 3: Common PHY Registers */
00231
00232 #define PHY_BCR              ((uint16_t)0x0000U)
00233 #define PHY_BSR              ((uint16_t)0x0001U)
00235 #define PHY_RESET             ((uint16_t)0x8000U)
00236 #define PHY_LOOPBACK          ((uint16_t)0x4000U)
00237 #define PHY_FULLDUPLEX_100M   ((uint16_t)0x2100U)
00238 #define PHY_HALFDUPLEX_100M   ((uint16_t)0x2000U)
00239 #define PHY_FULLDUPLEX_10M    ((uint16_t)0x0100U)
00240 #define PHY_HALFDUPLEX_10M    ((uint16_t)0x0000U)
00241 #define PHY_AUTONEGOTIATION   ((uint16_t)0x1000U)
00242 #define PHY_RESTART_AUTONEGOTIATION ((uint16_t)0x0200U)
00243 #define PHY_POWERDOWN         ((uint16_t)0x0800U)
00244 #define PHY_ISOLATE           ((uint16_t)0x0400U)
00246 #define PHY_AUTONEGO_COMPLETE ((uint16_t)0x0020U)
00247 #define PHY_LINKED_STATUS     ((uint16_t)0x0004U)
00248 #define PHY_JABBER_DETECTION ((uint16_t)0x0002U)
00250 /* Section 4: Extended PHY Registers */
00251 #define PHY_SR                ((uint16_t))
00253 #define PHY_SPEED_STATUS      ((uint16_t))
00254 #define PHY_DUPLEX_STATUS     ((uint16_t))
00256 /* ##### SPI peripheral configuration ###### */
00257
00258 /* CRC FEATURE: Use to activate CRC feature inside HAL SPI Driver
00259 * Activated: CRC code is present inside driver
00260 * Deactivated: CRC code cleaned from driver
00261 */
00262
00263 #define USE_SPI_CRC           0U
00264
00265 /* Includes -----*/
00270 #ifdef HAL_RCC_MODULE_ENABLED
00271   #include "stm32f7xx_hal_rcc.h"
00272 #endif /* HAL_RCC_MODULE_ENABLED */
00273
00274 #ifdef HAL_EXTI_MODULE_ENABLED
00275   #include "stm32f7xx_hal_exti.h"
00276 #endif /* HAL_EXTI_MODULE_ENABLED */
00277
00278 #ifdef HAL_GPIO_MODULE_ENABLED
00279   #include "stm32f7xx_hal_gpio.h"
00280 #endif /* HAL_GPIO_MODULE_ENABLED */
00281
00282 #ifdef HAL_DMA_MODULE_ENABLED
00283   #include "stm32f7xx_hal_dma.h"
00284 #endif /* HAL_DMA_MODULE_ENABLED */
00285
00286 #ifdef HAL_CORTEX_MODULE_ENABLED
00287   #include "stm32f7xx_hal_cortex.h"
00288 #endif /* HAL_CORTEX_MODULE_ENABLED */
00289
00290 #ifdef HAL_ADC_MODULE_ENABLED
00291   #include "stm32f7xx_hal_adc.h"

```

```
00292 #endif /* HAL_ADC_MODULE_ENABLED */
00293
00294 #ifdef HAL_CAN_MODULE_ENABLED
00295     #include "stm32f7xx_hal_can.h"
00296 #endif /* HAL_CAN_MODULE_ENABLED */
00297
00298 #ifdef HAL_CEC_MODULE_ENABLED
00299     #include "stm32f7xx_hal_cec.h"
00300 #endif /* HAL_CEC_MODULE_ENABLED */
00301
00302 #ifdef HAL_CRC_MODULE_ENABLED
00303     #include "stm32f7xx_hal_crc.h"
00304 #endif /* HAL_CRC_MODULE_ENABLED */
00305
00306 #ifdef HAL_CRYP_MODULE_ENABLED
00307     #include "stm32f7xx_hal_cryp.h"
00308 #endif /* HAL_CRYP_MODULE_ENABLED */
00309
00310 #ifdef HAL_DMA2D_MODULE_ENABLED
00311     #include "stm32f7xx_hal_dma2d.h"
00312 #endif /* HAL_DMA2D_MODULE_ENABLED */
00313
00314 #ifdef HAL_DAC_MODULE_ENABLED
00315     #include "stm32f7xx_hal_dac.h"
00316 #endif /* HAL_DAC_MODULE_ENABLED */
00317
00318 #ifdef HAL_DCMI_MODULE_ENABLED
00319     #include "stm32f7xx_hal_dcmi.h"
00320 #endif /* HAL_DCMI_MODULE_ENABLED */
00321
00322 #ifdef HAL_ETH_MODULE_ENABLED
00323     #include "stm32f7xx_hal_eth.h"
00324 #endif /* HAL_ETH_MODULE_ENABLED */
00325
00326 #ifdef HAL_ETH_LEGACY_MODULE_ENABLED
00327     #include "stm32f7xx_hal_eth_legacy.h"
00328 #endif /* HAL_ETH_LEGACY_MODULE_ENABLED */
00329
00330 #ifdef HAL_FLASH_MODULE_ENABLED
00331     #include "stm32f7xx_hal_flash.h"
00332 #endif /* HAL_FLASH_MODULE_ENABLED */
00333
00334 #ifdef HAL_SRAM_MODULE_ENABLED
00335     #include "stm32f7xx_hal_sram.h"
00336 #endif /* HAL_SRAM_MODULE_ENABLED */
00337
00338 #ifdef HAL_NOR_MODULE_ENABLED
00339     #include "stm32f7xx_hal_nor.h"
00340 #endif /* HAL_NOR_MODULE_ENABLED */
00341
00342 #ifdef HAL_NAND_MODULE_ENABLED
00343     #include "stm32f7xx_hal_nand.h"
00344 #endif /* HAL_NAND_MODULE_ENABLED */
00345
00346 #ifdef HAL_SDRAM_MODULE_ENABLED
00347     #include "stm32f7xx_hal_sdram.h"
00348 #endif /* HAL_SDRAM_MODULE_ENABLED */
00349
00350 #ifdef HAL_HASH_MODULE_ENABLED
00351     #include "stm32f7xx_hal_hash.h"
00352 #endif /* HAL_HASH_MODULE_ENABLED */
00353
00354 #ifdef HAL_I2C_MODULE_ENABLED
00355     #include "stm32f7xx_hal_i2c.h"
00356 #endif /* HAL_I2C_MODULE_ENABLED */
00357
00358 #ifdef HAL_I2S_MODULE_ENABLED
00359     #include "stm32f7xx_hal_i2s.h"
00360 #endif /* HAL_I2S_MODULE_ENABLED */
00361
00362 #ifdef HAL_IWDG_MODULE_ENABLED
00363     #include "stm32f7xx_hal_iwdg.h"
00364 #endif /* HAL_IWDG_MODULE_ENABLED */
00365
00366 #ifdef HAL_LPTIM_MODULE_ENABLED
00367     #include "stm32f7xx_hal_lptim.h"
00368 #endif /* HAL_LPTIM_MODULE_ENABLED */
00369
00370 #ifdef HAL_LTDC_MODULE_ENABLED
00371     #include "stm32f7xx_hal_ltddc.h"
00372 #endif /* HAL_LTDC_MODULE_ENABLED */
00373
00374 #ifdef HAL_PWR_MODULE_ENABLED
00375     #include "stm32f7xx_hal_pwr.h"
00376 #endif /* HAL_PWR_MODULE_ENABLED */
00377
00378 #ifdef HAL_QSPI_MODULE_ENABLED
```

```
00379 #include "stm32f7xx_hal_qspi.h"
00380 #endif /* HAL_QSPI_MODULE_ENABLED */
00381
00382 #ifdef HAL_RNG_MODULE_ENABLED
00383 #include "stm32f7xx_hal_rng.h"
00384 #endif /* HAL_RNG_MODULE_ENABLED */
00385
00386 #ifdef HAL_RTC_MODULE_ENABLED
00387 #include "stm32f7xx_hal_rtc.h"
00388 #endif /* HAL_RTC_MODULE_ENABLED */
00389
00390 #ifdef HAL_SAI_MODULE_ENABLED
00391 #include "stm32f7xx_hal_sai.h"
00392 #endif /* HAL_SAI_MODULE_ENABLED */
00393
00394 #ifdef HAL_SD_MODULE_ENABLED
00395 #include "stm32f7xx_hal_sd.h"
00396 #endif /* HAL_SD_MODULE_ENABLED */
00397
00398 #ifdef HAL_MMC_MODULE_ENABLED
00399 #include "stm32f7xx_hal_mmc.h"
00400 #endif /* HAL_MMC_MODULE_ENABLED */
00401
00402 #ifdef HAL_SPDIFRX_MODULE_ENABLED
00403 #include "stm32f7xx_hal_spdifrx.h"
00404 #endif /* HAL_SPDIFRX_MODULE_ENABLED */
00405
00406 #ifdef HAL_SPI_MODULE_ENABLED
00407 #include "stm32f7xx_hal_spi.h"
00408 #endif /* HAL_SPI_MODULE_ENABLED */
00409
00410 #ifdef HAL_TIM_MODULE_ENABLED
00411 #include "stm32f7xx_hal_tim.h"
00412 #endif /* HAL_TIM_MODULE_ENABLED */
00413
00414 #ifdef HAL_UART_MODULE_ENABLED
00415 #include "stm32f7xx_hal_uart.h"
00416 #endif /* HAL_UART_MODULE_ENABLED */
00417
00418 #ifdef HAL_USART_MODULE_ENABLED
00419 #include "stm32f7xx_hal_usart.h"
00420 #endif /* HAL_USART_MODULE_ENABLED */
00421
00422 #ifdef HAL_IRDA_MODULE_ENABLED
00423 #include "stm32f7xx_hal_irda.h"
00424 #endif /* HAL_IRDA_MODULE_ENABLED */
00425
00426 #ifdef HAL_SMARTCARD_MODULE_ENABLED
00427 #include "stm32f7xx_hal_smartcard.h"
00428 #endif /* HAL_SMARTCARD_MODULE_ENABLED */
00429
00430 #ifdef HAL_WWDG_MODULE_ENABLED
00431 #include "stm32f7xx_hal_wwdg.h"
00432 #endif /* HAL_WWDG_MODULE_ENABLED */
00433
00434 #ifdef HAL_PCD_MODULE_ENABLED
00435 #include "stm32f7xx_hal_pcd.h"
00436 #endif /* HAL_PCD_MODULE_ENABLED */
00437
00438 #ifdef HAL_HCD_MODULE_ENABLED
00439 #include "stm32f7xx_hal_hcd.h"
00440 #endif /* HAL_HCD_MODULE_ENABLED */
00441
00442 #ifdef HAL_DFSDM_MODULE_ENABLED
00443 #include "stm32f7xx_hal_dfsm.h"
00444 #endif /* HAL_DFSDM_MODULE_ENABLED */
00445
00446 #ifdef HAL_DSI_MODULE_ENABLED
00447 #include "stm32f7xx_hal_dsi.h"
00448 #endif /* HAL_DSI_MODULE_ENABLED */
00449
00450 #ifdef HAL_JPEG_MODULE_ENABLED
00451 #include "stm32f7xx_hal_jpeg.h"
00452 #endif /* HAL_JPEG_MODULE_ENABLED */
00453
00454 #ifdef HAL_MDIOS_MODULE_ENABLED
00455 #include "stm32f7xx_hal_mdios.h"
00456 #endif /* HAL_MDIOS_MODULE_ENABLED */
00457
00458 #ifdef HAL_SMBUS_MODULE_ENABLED
00459 #include "stm32f7xx_hal_smbus.h"
00460 #endif /* HAL_SMBUS_MODULE_ENABLED */
00461
00462 /* Exported macro -----*/
00463 #ifdef USE_FULL_ASSERT
00464 #define assert_param(expr) ((expr) ? (void)0U : assert_failed((uint8_t *)__FILE__, __LINE__))
00465 /* Exported functions -----*/
```

```

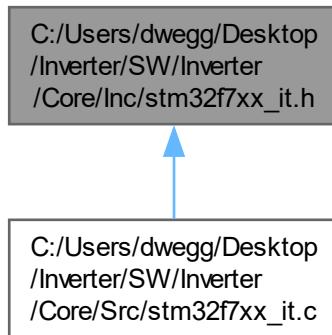
00474     void assert_failed(uint8_t* file, uint32_t line);
00475 #else
00476     #define assert_param(expr) ((void)0U)
00477 #endif /* USE_FULL_ASSERT */
00478
00479 #ifdef __cplusplus
00480 }
00481 #endif
00482
00483 #endif /* __STM32F7xx_HAL_CONF_H */
00484

```

## 6.31 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/stm32f7xx\_it.h File Reference

This file contains the headers of the interrupt handlers.

This graph shows which files directly or indirectly include this file:



### 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)

- void [SysTick\\_Handler](#) (void)  
*This function handles Pendable request for system service.*
- void [CAN1\\_RX0\\_IRQHandler](#) (void)  
*This function handles System tick timer.*
- void [CAN1\\_RX1\\_IRQHandler](#) (void)  
*This function handles CAN1 RX0 interrupts.*
- void [TIM1\\_UP\\_TIM10\\_IRQHandler](#) (void)  
*This function handles CAN1 RX1 interrupt.*
- void [TIM1\\_UP\\_TIM10\\_IRQHandler](#) (void)  
*This function handles TIM1 update interrupt and TIM10 global interrupt.*
- void [TIM6\\_DAC\\_IRQHandler](#) (void)  
*This function handles TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts.*
- void [DMA2\\_Stream1\\_IRQHandler](#) (void)  
*This function handles DMA2 stream1 global interrupt.*
- void [DMA2\\_Stream2\\_IRQHandler](#) (void)  
*This function handles DMA2 stream2 global interrupt.*

### 6.31.1 Detailed Description

This file contains the headers of the interrupt handlers.

#### Attention

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

### 6.31.2 Function Documentation

#### 6.31.2.1 BusFault\_Handler()

```
void BusFault_Handler (
    void )
```

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

#### 6.31.2.2 CAN1\_RX0\_IRQHandler()

```
void CAN1_RX0_IRQHandler (
    void )
```

This function handles CAN1 RX0 interrupts.

### 6.31.2.3 CAN1\_RX1\_IRQHandler()

```
void CAN1_RX1_IRQHandler (
    void )
```

This function handles CAN1 RX1 interrupt.

### 6.31.2.4 DebugMon\_Handler()

```
void DebugMon_Handler (
    void )
```

This function handles Debug monitor.

### 6.31.2.5 DMA2\_Stream1\_IRQHandler()

```
void DMA2_Stream1_IRQHandler (
    void )
```

This function handles DMA2 stream1 global interrupt.

### 6.31.2.6 DMA2\_Stream2\_IRQHandler()

```
void DMA2_Stream2_IRQHandler (
    void )
```

This function handles DMA2 stream2 global interrupt.

### 6.31.2.7 HardFault\_Handler()

```
void HardFault_Handler (
    void )
```

This function handles Hard fault interrupt.

### 6.31.2.8 MemManage\_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

### 6.31.2.9 NMI\_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

### 6.31.2.10 PendSV\_Handler()

```
void PendSV_Handler (
    void )
```

This function handles Pendable request for system service.

### 6.31.2.11 SVC\_Handler()

```
void SVC_Handler (
    void )
```

This function handles System service call via SWI instruction.

### 6.31.2.12 SysTick\_Handler()

```
void SysTick_Handler (
    void )
```

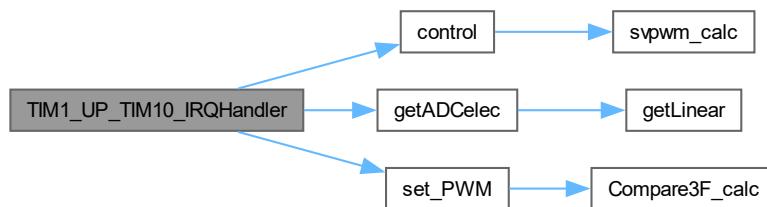
This function handles System tick timer.

### 6.31.2.13 TIM1\_UP\_TIM10\_IRQHandler()

```
void TIM1_UP_TIM10_IRQHandler (
    void )
```

This function handles TIM1 update interrupt and TIM10 global interrupt.

Here is the call graph for this function:



### 6.31.2.14 TIM6\_DAC\_IRQHandler()

```
void TIM6_DAC_IRQHandler (
    void )
```

This function handles TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts.

Here is the call graph for this function:



### 6.31.2.15 UsageFault\_Handler()

```
void UsageFault_Handler (
    void )
```

This function handles Undefined instruction or illegal state.

## 6.32 stm32f7xx\_it.h

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

```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020 /* Define to prevent recursive inclusion -----*/
00021 #ifndef __STM32F7XX_IT_H
00022 #define __STM32F7XX_IT_H
00023
00024 #ifdef __cplusplus
00025     extern "C" {
00026 #endif
00027
00028 /* Private includes -----*/
00029 /* USER CODE BEGIN Includes */
00030
00031 /* USER CODE END Includes */
00032
00033 /* Exported types -----*/
00034 /* USER CODE BEGIN ET */
00035
00036 /* USER CODE END ET */
00037
00038 /* Exported constants -----*/
00039 /* USER CODE BEGIN EC */
00040
00041 /* USER CODE END EC */
00042
00043 /* Exported macro -----*/
00044 /* USER CODE BEGIN EM */
00045
00046 /* USER CODE END EM */
00047
00048 /* Exported functions prototypes -----*/
00049 void NMI_Handler(void);
00050 void HardFault_Handler(void);
00051 void MemManage_Handler(void);
00052 void BusFault_Handler(void);
00053 void UsageFault_Handler(void);
```

```

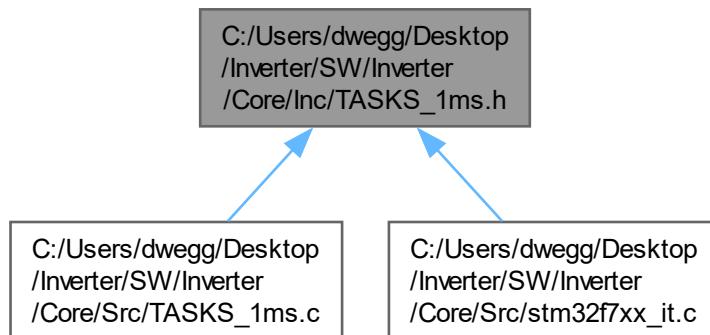
00054 void SVC_Handler(void);
00055 void DebugMon_Handler(void);
00056 void PendSV_Handler(void);
00057 void SysTick_Handler(void);
00058 void CAN1_RX0_IRQHandler(void);
00059 void CAN1_RX1_IRQHandler(void);
00060 void TIM1_UP_TIM10_IRQHandler(void);
00061 void TIM6_DAC_IRQHandler(void);
00062 void DMA2_Stream1_IRQHandler(void);
00063 void DMA2_Stream2_IRQHandler(void);
00064 /* USER CODE BEGIN EFP */
00065
00066 /* USER CODE END EFP */
00067
00068 #ifdef __cplusplus
00069 }
00070 #endif
00071
00072 #endif /* __STM32F7xx_IT_H */

```

## 6.33 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/TASKS\_1ms.h File Reference

Header file for functions related to tasks executed every 1ms.

This graph shows which files directly or indirectly include this file:



### Functions

- void `tasks_1ms` (void)  
*Function to be executed every 1ms.*

#### 6.33.1 Detailed Description

Header file for functions related to tasks executed every 1ms.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg in GitHub). All rights reserved.

This software is licensed under the Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. For more information, see: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

## 6.33.2 Function Documentation

### 6.33.2.1 tasks\_1ms()

```
void tasks_1ms (
    void )
```

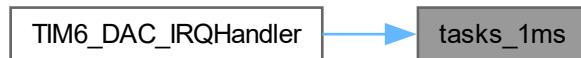
Function to be executed every 1ms.

This function is called by the TIM6 IRQ handler every millisecond.

This function is called by the TIM6 IRQ handler every millisecond. It increments the millisecond counter and calls the [LED](#) handler for left, right, and error LEDs. Here is the call graph for this function:



Here is the caller graph for this function:



## 6.34 TASKS\_1ms.h

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

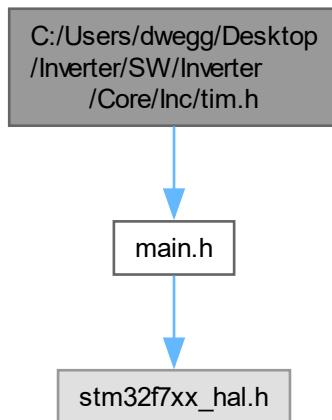
```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020
00021 #ifndef TASKS_1MS_H
00022 #define TASKS_1MS_H
00023
00029 void tasks_1ms(void);
00030
00031 #endif /* TASKS_1MS_H */
```

## 6.35 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/tim.h File Reference

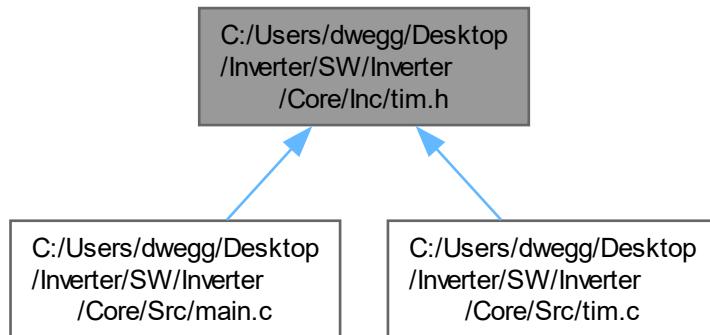
This file contains all the function prototypes for the [tim.c](#) file.

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

Include dependency graph for tim.h:



This graph shows which files directly or indirectly include this file:



### Functions

- void [MX\\_TIM1\\_Init](#) (void)
- void [MX\\_TIM2\\_Init](#) (void)
- void [MX\\_TIM4\\_Init](#) (void)
- void [MX\\_TIM6\\_Init](#) (void)
- void [MX\\_TIM8\\_Init](#) (void)
- void [HAL\\_TIM\\_MspPostInit](#) (TIM\_HandleTypeDef \*htim)

## Variables

- TIM\_HandleTypeDef [htim1](#)
- TIM\_HandleTypeDef [htim2](#)
- TIM\_HandleTypeDef [htim4](#)
- TIM\_HandleTypeDef [htim6](#)
- TIM\_HandleTypeDef [htim8](#)

### 6.35.1 Detailed Description

This file contains all the function prototypes for the [tim.c](#) file.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

### 6.35.2 Function Documentation

#### 6.35.2.1 HAL\_TIM\_MspPostInit()

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

TIM1 GPIO Configuration PE8 -----> TIM1\_CH1N PE9 -----> TIM1\_CH1 PE10 -----> TIM1\_CH2N PE11 ----->  
TIM1\_CH2 PE12 -----> TIM1\_CH3N PE13 -----> TIM1\_CH3

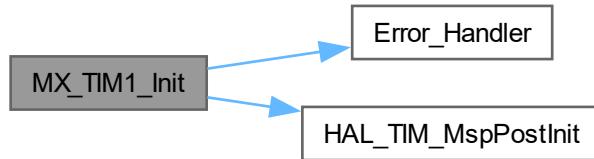
TIM8 GPIO Configuration PA5 -----> TIM8\_CH1N PB14 -----> TIM8\_CH2N PB15 -----> TIM8\_CH3N PC6 ----->  
TIM8\_CH1 PC7 -----> TIM8\_CH2 PC8 -----> TIM8\_CH3Here is the caller graph for this function:



### 6.35.2.2 MX\_TIM1\_Init()

```
void MX_TIM1_Init (
    void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.35.2.3 MX\_TIM2\_Init()

```
void MX_TIM2_Init (
    void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.35.2.4 MX\_TIM4\_Init()

```
void MX_TIM4_Init (
    void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.35.2.5 MX\_TIM6\_Init()

```
void MX_TIM6_Init (
    void  )
```

Here is the call graph for this function:



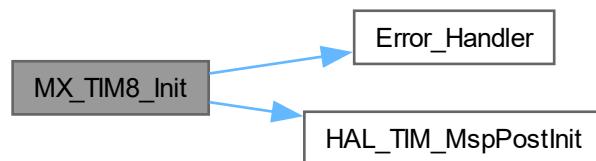
Here is the caller graph for this function:



### 6.35.2.6 MX\_TIM8\_Init()

```
void MX_TIM8_Init (
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.35.3 Variable Documentation

#### 6.35.3.1 htim1

```
TIM_HandleTypeDef htim1 [extern]
```

#### 6.35.3.2 htim2

```
TIM_HandleTypeDef htim2 [extern]
```

#### 6.35.3.3 htim4

```
TIM_HandleTypeDef htim4 [extern]
```

#### 6.35.3.4 htim6

```
TIM_HandleTypeDef htim6 [extern]
```

#### 6.35.3.5 htim8

```
TIM_HandleTypeDef htim8 [extern]
```

## 6.36 tim.h

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

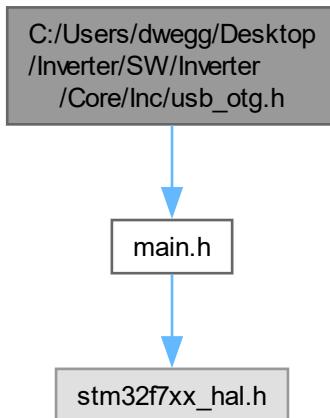
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -----*/
00021 #ifndef __TIM_H__
00022 #define __TIM_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes -----*/
00029 #include "main.h"
00030
00031 /* USER CODE BEGIN Includes */
00032
00033 /* USER CODE END Includes */
00034
00035 extern TIM_HandleTypeDef htim1;
00036
00037 extern TIM_HandleTypeDef htim2;
00038
00039 extern TIM_HandleTypeDef htim4;
00040
00041 extern TIM_HandleTypeDef htim6;
00042
00043 extern TIM_HandleTypeDef htim8;
00044
00045 /* USER CODE BEGIN Private defines */
00046
00047 /* USER CODE END Private defines */
00048
00049 void MX_TIM1_Init(void);
00050 void MX_TIM2_Init(void);
00051 void MX_TIM4_Init(void);
```

```
00052 void MX_TIM6_Init(void);
00053 void MX_TIM8_Init(void);
00054
00055 void HAL_TIM_MspPostInit(TIM_HandleTypeDef *htim);
00056
00057 /* USER CODE BEGIN Prototypes */
00058
00059 /* USER CODE END Prototypes */
00060
00061 #ifdef __cplusplus
00062 }
00063 #endif
00064
00065 #endif /* __TIM_H__ */
00066
```

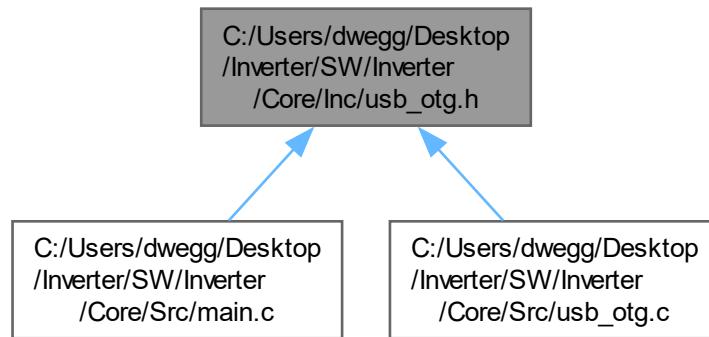
## 6.37 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/usb\_otg.h File Reference

This file contains all the function prototypes for the [usb\\_otg.c](#) file.

```
#include "main.h"
Include dependency graph for usb_otg.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- void [MX\\_USB\\_OTG\\_FS\\_USB\\_Init](#) (void)

### 6.37.1 Detailed Description

This file contains all the function prototypes for the [usb\\_otg.c](#) file.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

### 6.37.2 Function Documentation

#### 6.37.2.1 MX\_USB\_OTG\_FS\_USB\_Init()

```
void MX_USB_OTG_FS_USB_Init (
    void )
```

Here is the caller graph for this function:



## 6.38 usb\_otg.h

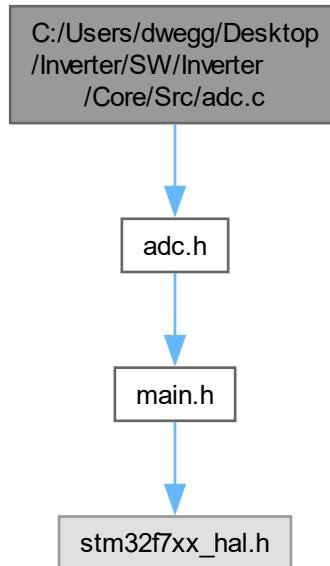
[Go to the documentation of this file.](#)

```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -----*/
00021 #ifndef __USB_OTG_H__
00022 #define __USB_OTG_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes -----*/
00029 #include "main.h"
00030
00031 /* USER CODE BEGIN Includes */
00032
00033 /* USER CODE END Includes */
00034
00035 /* USER CODE BEGIN Private defines */
00036
00037 /* USER CODE END Private defines */
00038
00039 void MX_USB_OTG_FS_USB_Init(void);
00040
00041 /* USER CODE BEGIN Prototypes */
00042
00043 /* USER CODE END Prototypes */
00044
00045 #ifdef __cplusplus
00046 }
00047 #endif
00048
00049 #endif /* __USB_OTG_H__ */
00050
```

## 6.39 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/adc.c File Reference

This file provides code for the configuration of the ADC instances.

```
#include "adc.h"
Include dependency graph for adc.c:
```



## Functions

- void [MX\\_ADC1\\_Init](#) (void)
- void [MX\\_ADC2\\_Init](#) (void)
- void [MX\\_ADC3\\_Init](#) (void)
- void [HAL\\_ADC\\_MspInit](#) (ADC\_HandleTypeDef \*adcHandle)
- void [HAL\\_ADC\\_MspDeInit](#) (ADC\_HandleTypeDef \*adcHandle)

## Variables

- ADC\_HandleTypeDef [hadc1](#)
- ADC\_HandleTypeDef [hadc2](#)
- ADC\_HandleTypeDef [hadc3](#)
- DMA\_HandleTypeDef [hdma\\_adc2](#)
- DMA\_HandleTypeDef [hdma\\_adc3](#)

### 6.39.1 Detailed Description

This file provides code for the configuration of the ADC instances.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

## 6.39.2 Function Documentation

### 6.39.2.1 HAL\_ADC\_MspDeInit()

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

ADC1 GPIO Configuration PA0/WKUP -----> ADC1\_IN0 PA1 -----> ADC1\_IN1 PA2 -----> ADC1\_IN2 PA3 -----> ADC1\_IN3

ADC2 GPIO Configuration PA6 -----> ADC2\_IN6 PA7 -----> ADC2\_IN7 PB0 -----> ADC2\_IN8 PB1 -----> ADC2\_IN9

ADC3 GPIO Configuration PC0 -----> ADC3\_IN10 PC1 -----> ADC3\_IN11 PC2 -----> ADC3\_IN12 PC3 -----> ADC3\_IN13

### 6.39.2.2 HAL\_ADC\_MspInit()

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

ADC1 GPIO Configuration PA0/WKUP -----> ADC1\_IN0 PA1 -----> ADC1\_IN1 PA2 -----> ADC1\_IN2 PA3 -----> ADC1\_IN3

ADC2 GPIO Configuration PA6 -----> ADC2\_IN6 PA7 -----> ADC2\_IN7 PB0 -----> ADC2\_IN8 PB1 -----> ADC2\_IN9

ADC3 GPIO Configuration PC0 -----> ADC3\_IN10 PC1 -----> ADC3\_IN11 PC2 -----> ADC3\_IN12 PC3 -----> ADC3\_IN13

Here is the call graph for this function:



### 6.39.2.3 MX\_ADC1\_Init()

```
void MX_ADC1_Init (
    void )
```

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

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

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

Configure for the selected ADC regular channel its corresponding rank in the sequencer and its sample time. Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.39.2.4 MX\_ADC2\_Init()

```
void MX_ADC2_Init (   
    void )
```

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

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

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

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

Configure for the selected ADC regular channel its corresponding rank in the sequencer and its sample time. Here is the call graph for this function:



Here is the caller graph for this function:



### 6.39.2.5 MX\_ADC3\_Init()

```
void MX_ADC3_Init (
    void )
```

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

Configure for the selected ADC regular channel its corresponding rank in the sequencer and its sample time. Here is the call graph for this function:



Here is the caller graph for this function:



## 6.39.3 Variable Documentation

### 6.39.3.1 hadc1

```
ADC_HandleTypeDef hadc1
```

### 6.39.3.2 hadc2

```
ADC_HandleTypeDef hadc2
```

### 6.39.3.3 hadc3

```
ADC_HandleTypeDef hadc3
```

### 6.39.3.4 hdma\_adc2

```
DMA_HandleTypeDef hdma_adc2
```

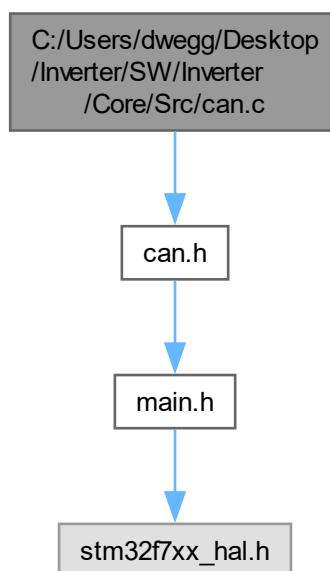
### 6.39.3.5 hdma\_adc3

```
DMA_HandleTypeDef hdma_adc3
```

## 6.40 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/can.c File Reference

This file provides code for the configuration of the CAN instances.

```
#include "can.h"  
Include dependency graph for can.c:
```



## Functions

- void [MX\\_CAN1\\_Init](#) (void)
- void [HAL\\_CAN\\_MspInit](#) (CAN\_HandleTypeDef \*canHandle)
- void [HAL\\_CAN\\_MspDelInit](#) (CAN\_HandleTypeDef \*canHandle)

## Variables

- CAN\_HandleTypeDef [hcan1](#)

### 6.40.1 Detailed Description

This file provides code for the configuration of the CAN instances.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

### 6.40.2 Function Documentation

#### 6.40.2.1 HAL\_CAN\_MspDelInit()

```
void HAL_CAN_MspDeInit (
    CAN_HandleTypeDef * canHandle )
```

CAN1 GPIO Configuration PD0 -----> CAN1\_RX PD1 -----> CAN1\_TX

#### 6.40.2.2 HAL\_CAN\_MspInit()

```
void HAL_CAN_MspInit (
    CAN_HandleTypeDef * canHandle )
```

CAN1 GPIO Configuration PD0 -----> CAN1\_RX PD1 -----> CAN1\_TX

#### 6.40.2.3 MX\_CAN1\_Init()

```
void MX_CAN1_Init (
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.40.3 Variable Documentation

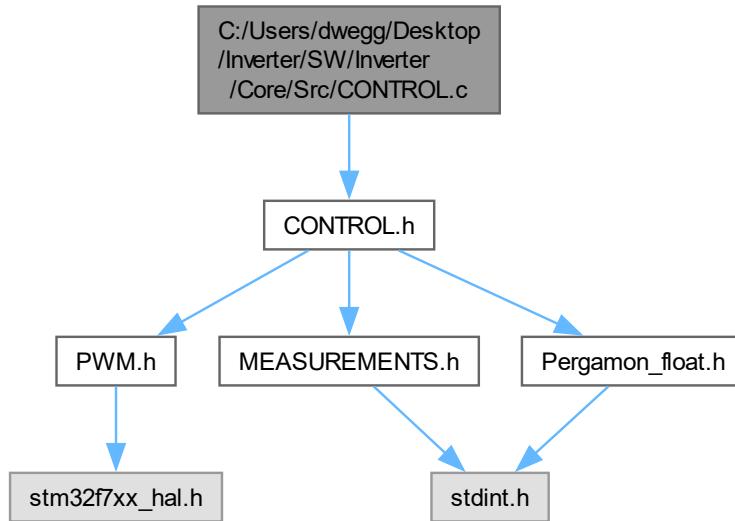
##### 6.40.3.1 hcan1

```
CAN_HandleTypeDef hcan1
```

### 6.41 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/CONTROL.c File Reference

Source file for control logic.

```
#include "CONTROL.h"
Include dependency graph for CONTROL.c:
```



## Functions

- void `control` (float `valfa`, float `vbeta`, `Measurements` `measurements`, volatile `Duties` \*`duties`)  
*Control function.*

## Variables

- float `valfa_L` = -0.2F
- float `vbeta_L` = 0.2F

### 6.41.1 Detailed Description

Source file for control logic.

### 6.41.2 Function Documentation

#### 6.41.2.1 `control()`

```
void control (
    float valfa,
    float vbeta,
    Measurements measurements,
    volatile Duties * duties )
```

Control function.

This function implements the control logic based on measurements. For now, it sets a constant duty cycle.

**Parameters**

<i>measurements</i>	Measurements structure.
<i>duties</i>	Pointer to the duties structure.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.41.3 Variable Documentation

#### 6.41.3.1 valfa\_L

```
float valfa_L = -0.2F
```

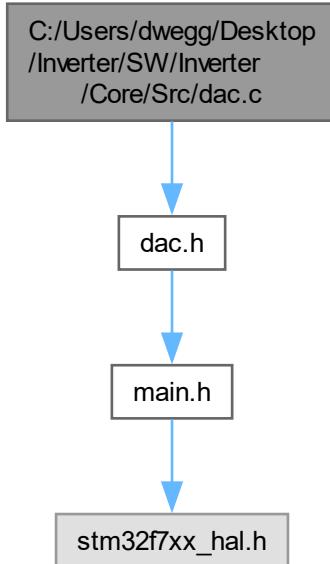
#### 6.41.3.2 vbeta\_L

```
float vbeta_L = 0.2F
```

## 6.42 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/dac.c File Reference

This file provides code for the configuration of the DAC instances.

```
#include "dac.h"  
Include dependency graph for dac.c:
```



## Functions

- void [MX\\_DAC\\_Init](#) (void)
- void [HAL\\_DAC\\_MspInit](#) (DAC\_HandleTypeDef \*dacHandle)
- void [HAL\\_DAC\\_MspDeInit](#) (DAC\_HandleTypeDef \*dacHandle)

## Variables

- DAC\_HandleTypeDef [hdac](#)

### 6.42.1 Detailed Description

This file provides code for the configuration of the DAC instances.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

## 6.42.2 Function Documentation

### 6.42.2.1 HAL\_DAC\_MspDeInit()

```
void HAL_DAC_MspDeInit (
    DAC_HandleTypeDef * dacHandle )
```

DAC GPIO Configuration PA4 -----> DAC\_OUT1

Uncomment the line below to disable the "TIM6\_DAC\_IRQn" interrupt Be aware, disabling shared interrupt may affect other IPs

### 6.42.2.2 HAL\_DAC\_MspInit()

```
void HAL_DAC_MspInit (
    DAC_HandleTypeDef * dacHandle )
```

DAC GPIO Configuration PA4 -----> DAC\_OUT1

### 6.42.2.3 MX\_DAC\_Init()

```
void MX_DAC_Init (
    void )
```

DAC Initialization

DAC channel OUT1 configHere is the call graph for this function:



Here is the caller graph for this function:



### 6.42.3 Variable Documentation

#### 6.42.3.1 hdac

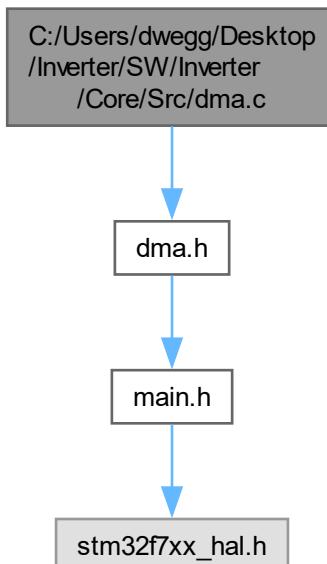
```
DAC_HandleTypeDef hdac
```

## 6.43 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/dma.c File Reference

This file provides code for the configuration of all the requested memory to memory DMA transfers.

```
#include "dma.h"
```

Include dependency graph for dma.c:



### Functions

- void [MX\\_DMA\\_Init](#) (void)

### 6.43.1 Detailed Description

This file provides code for the configuration of all the requested memory to memory DMA transfers.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

## 6.43.2 Function Documentation

### 6.43.2.1 MX\_DMA\_Init()

```
void MX_DMA_Init (
    void )
```

Enable DMA controller clock Here is the caller graph for this function:

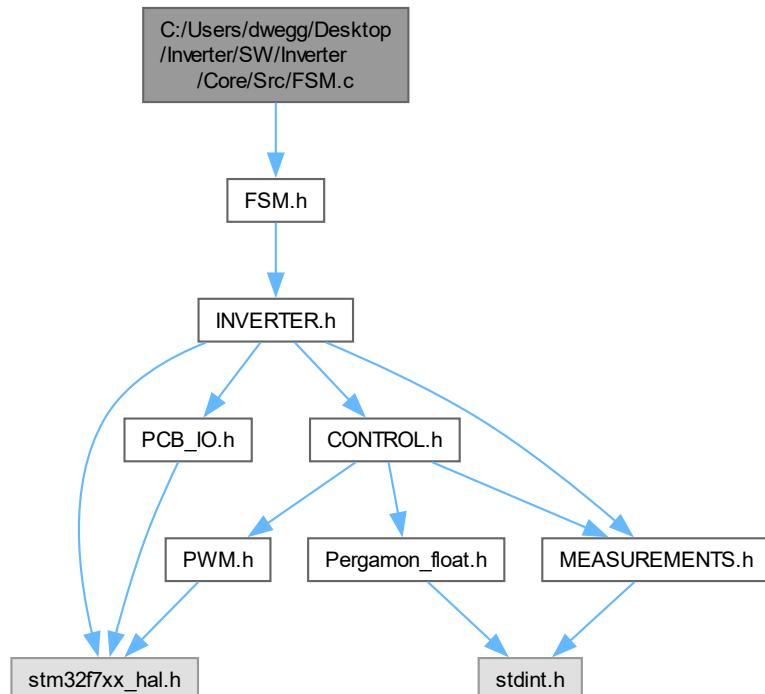


## 6.44 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/FSM.c File Reference

This file provides code for Finite State Machine (FSM) control.

```
#include "FSM.h"
```

Include dependency graph for FSM.c:



## Functions

- void `inv_FSM (inverterStruct *inv)`  
*Execute the finite state machine for inverter.*

### 6.44.1 Detailed Description

This file provides code for Finite State Machine (FSM) control.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg in GitHub). All rights reserved.

This software is licensed under the Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. For more information, see: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

### 6.44.2 Function Documentation

#### 6.44.2.1 `inv_FSM()`

```
void inv_FSM (
    inverterStruct * inv )
```

Execute the finite state machine for inverter.

Run the Finite State Machine (FSM) for inverter operation control.

This function executes the finite state machine to control the inverter based on its current state.

#### Parameters

<code>inv</code>	Pointer to the inverter structure.
------------------	------------------------------------

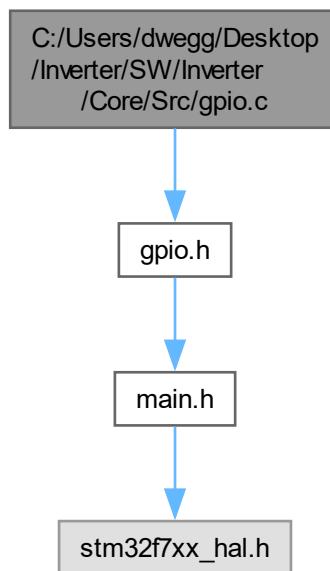
Here is the caller graph for this function:



## 6.45 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/gpio.c File Reference

This file provides code for the configuration of all used GPIO pins.

```
#include "gpio.h"  
Include dependency graph for gpio.c:
```



### Functions

- void [MX\\_GPIO\\_Init](#) (void)

#### 6.45.1 Detailed Description

This file provides code for the configuration of all used GPIO pins.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

## 6.45.2 Function Documentation

### 6.45.2.1 MX\_GPIO\_Init()

```
void MX_GPIO_Init (
    void )
```

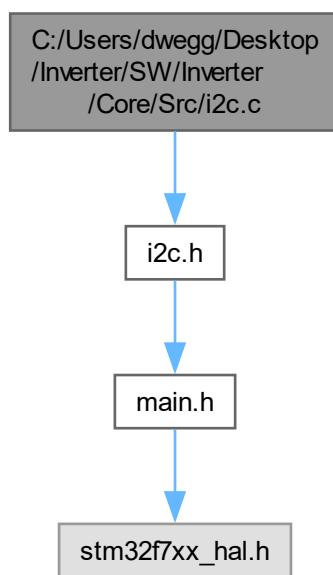
Configure pins as Analog Input Output EVENT\_OUT EXTI PA9 -----> USB\_OTG\_FS\_VBUS PA10 -----> USB↔\_OTG\_FS\_ID PA11 -----> USB\_OTG\_FS\_DM PA12 -----> USB\_OTG\_FS\_DP Here is the caller graph for this function:



## 6.46 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/i2c.c File Reference

This file provides code for the configuration of the I2C instances.

```
#include "i2c.h"
Include dependency graph for i2c.c:
```



## Functions

- void [MX\\_I2C1\\_Init](#) (void)
- void [HAL\\_I2C\\_MspInit](#) (I2C\_HandleTypeDef \*i2cHandle)
- void [HAL\\_I2C\\_MspDelInit](#) (I2C\_HandleTypeDef \*i2cHandle)

## Variables

- I2C\_HandleTypeDef [hi2c1](#)

### 6.46.1 Detailed Description

This file provides code for the configuration of the I2C instances.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

### 6.46.2 Function Documentation

#### 6.46.2.1 HAL\_I2C\_MspDelInit()

```
void HAL_I2C_MspDeInit (
    I2C_HandleTypeDef * i2cHandle )
```

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

#### 6.46.2.2 HAL\_I2C\_MspInit()

```
void HAL_I2C_MspInit (
    I2C_HandleTypeDef * i2cHandle )
```

Initializes the peripherals clock

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

### 6.46.2.3 MX\_I2C1\_Init()

```
void MX_I2C1_Init (
    void )
```

Configure Analogue filter

Configure Digital filterHere is the call graph for this function:



Here is the caller graph for this function:



## 6.46.3 Variable Documentation

### 6.46.3.1 hi2c1

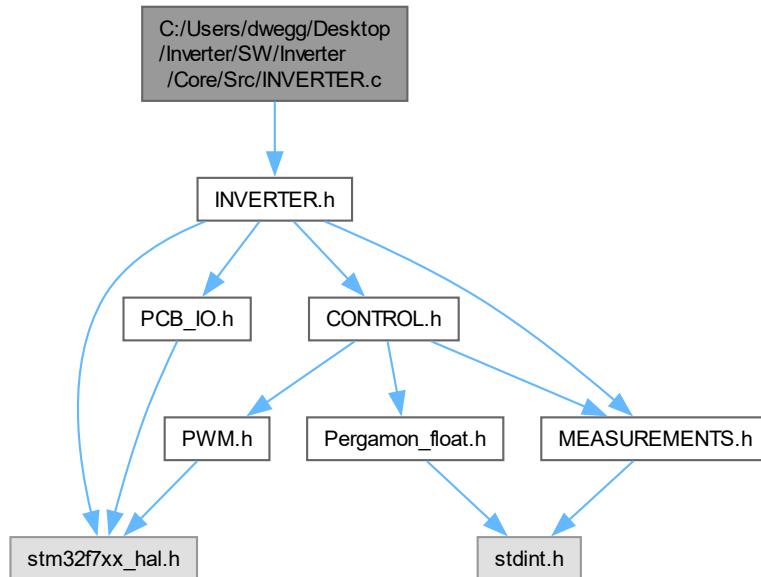
```
I2C_HandleTypeDef hi2c1
```

## 6.47 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/INVERTER.c File Reference

This file provides code for Finite State Machine (FSM) control.

```
#include "INVERTER.h"
```

Include dependency graph for INVERTER.c:



## Functions

- void `inv_init (inverterStruct *inv, LED *led, GPIO_TypeDef *enable_port, uint16_t enable_pin, TIM_HandleTypeDef *htim, ADC_HandleTypeDef *hadc)`

*Initialize the inverter.*

## Variables

- `inverterStruct invLeft = {0}`
- `inverterStruct invRight = {0}`

### 6.47.1 Detailed Description

This file provides code for Finite State Machine (FSM) control.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg in GitHub). All rights reserved.

This software is licensed under the Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. For more information, see: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

## 6.47.2 Function Documentation

### 6.47.2.1 inv\_init()

```
void inv_init (
    inverterStruct * inv,
    LED * led,
    GPIO_TypeDef * enable_port,
    uint16_t enable_pin,
    TIM_HandleTypeDef * htim,
    ADC_HandleTypeDef * hadc )
```

Initialize the inverter.

This function initializes the inverter structure with the specified [LED](#), GPIO port, and pin.

#### Parameters

<i>inv</i>	Pointer to the inverter structure.
<i>led</i>	Pointer to the <a href="#">LED</a> structure.
<i>enable_port</i>	Pointer to the GPIO port for enabling/disabling the inverter.
<i>enable_pin</i>	Pin number for enabling/disabling the inverter.
<i>htim</i>	Timer peripheral for the PWM output.
<i>hadc</i>	ADC peripheral for the current phase current and DC voltage sensing.

Here is the call graph for this function:



Here is the caller graph for this function:



## 6.47.3 Variable Documentation

### 6.47.3.1 invLeft

```
inverterStruct invLeft = {0}
```

### 6.47.3.2 invRight

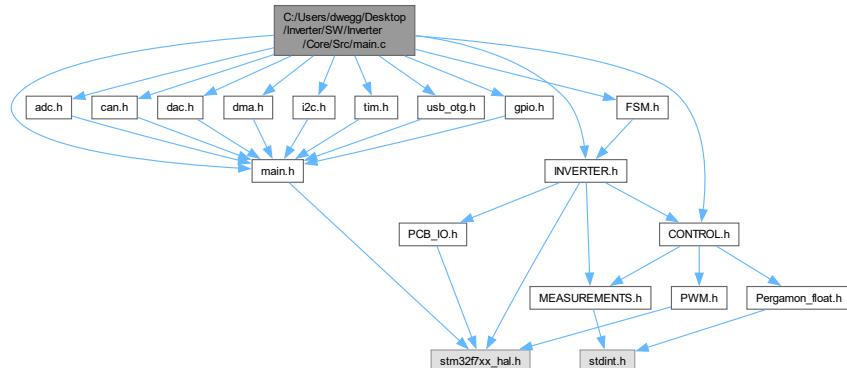
```
inverterStruct invRight = {0}
```

## 6.48 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/main.c File Reference

: Main program body

```
#include "main.h"
#include "adc.h"
#include "can.h"
#include "dac.h"
#include "dma.h"
#include "i2c.h"
#include "tim.h"
#include "usb_otg.h"
#include "gpio.h"
#include "FSM.h"
#include "CONTROL.h"
#include "INVERTER.h"
```

Include dependency graph for main.c:



## 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.*

## 6.48.1 Detailed Description

: Main program body

### Attention

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

## 6.48.2 Function Documentation

### 6.48.2.1 Error\_Handler()

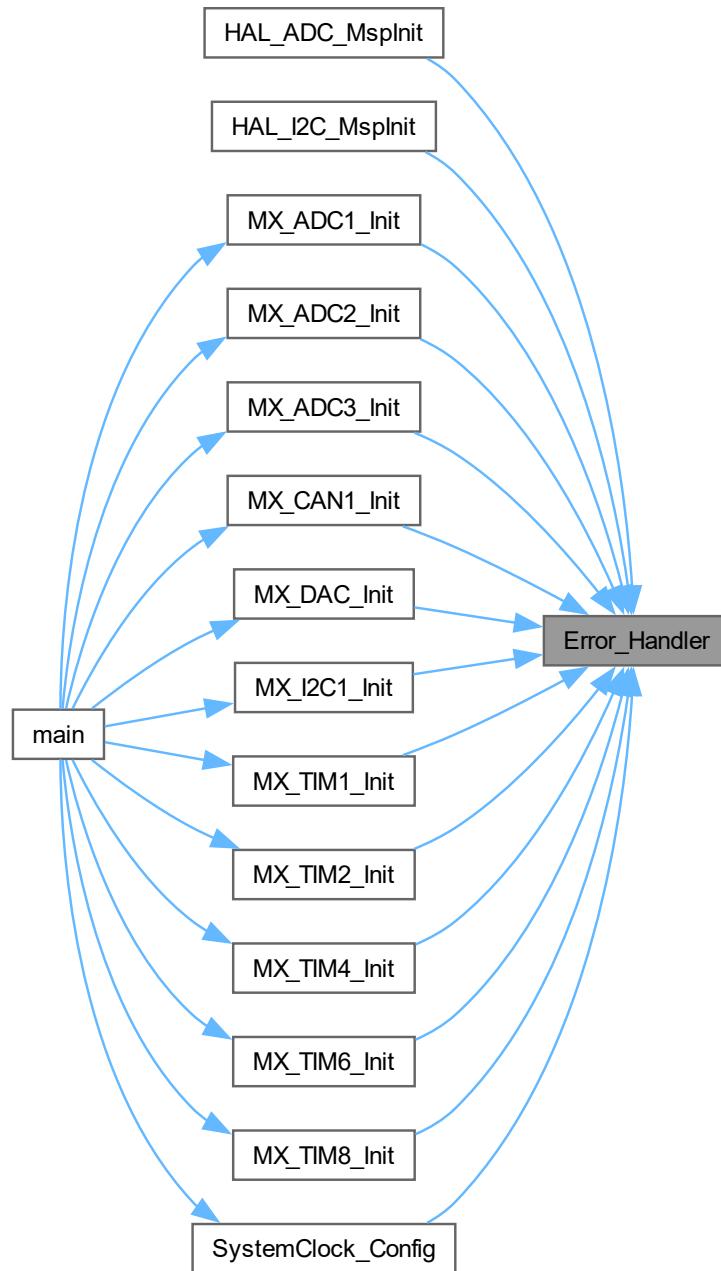
```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

#### Return values

None	<input type="button" value=""/>
------	---------------------------------

Here is the caller graph for this function:



#### 6.48.2.2 main()

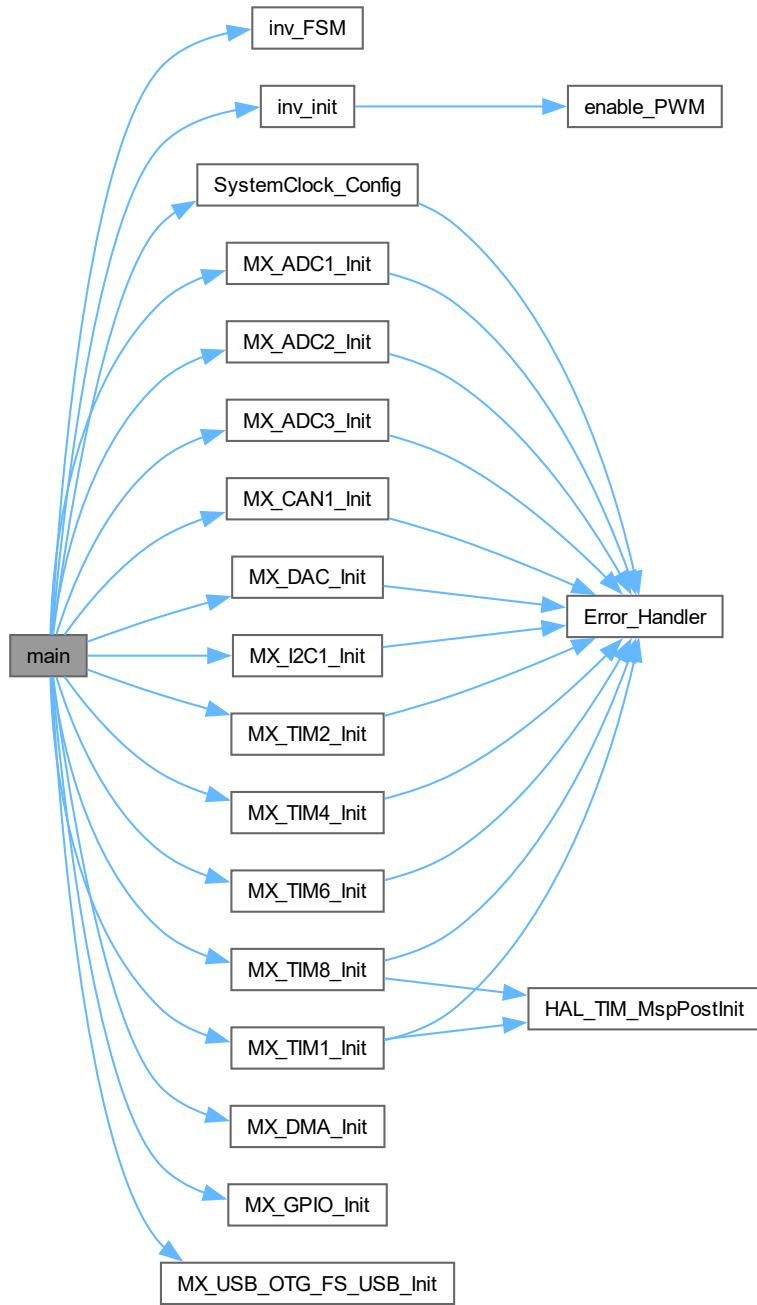
```
int main (
    void )
```

The application entry point.

Return values

int	
-----	--

Here is the call graph for this function:



#### 6.48.2.3 SystemClock\_Config()

```
void SystemClock_Config (
```

```
void  )
```

System Clock Configuration.

Return values

None	
------	--

Configure the main internal regulator output voltage

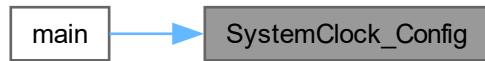
Initializes the RCC Oscillators according to the specified parameters in the RCC\_OsclInitTypeDef structure.

Activate the Over-Drive mode

Initializes the CPU, AHB and APB buses clocksHere is the call graph for this function:



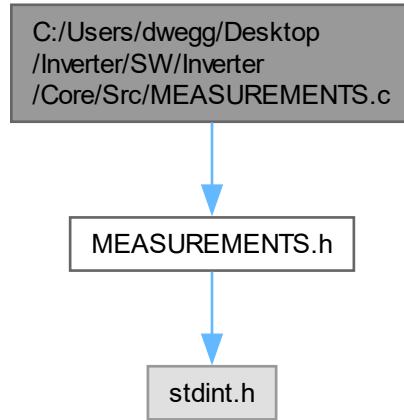
Here is the caller graph for this function:



## 6.49 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/← MEASUREMENTS.c File Reference

This file provides functions for handling measurements.

```
#include "MEASUREMENTS.h"
Include dependency graph for MEASUREMENTS.c:
```



## Functions

- `uint8_t getADCelec (volatile uint32_t *ADC_raw, volatile Measurements *measurements)`  
*Get electrical ADC measurements.*
- `float getLinear (uint32_t bits, float slope, float offset)`  
*Convert ADC reading to physical measurement with linear response.*

### 6.49.1 Detailed Description

This file provides functions for handling measurements.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg in GitHub). All rights reserved.

This software is licensed under the Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. For more information, see: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

### 6.49.2 Function Documentation

#### 6.49.2.1 getADCelec()

```
uint8_t getADCelec (
    volatile uint32_t * ADC_raw,
    volatile Measurements * measurements )
```

Get electrical ADC measurements.

**Parameters**

<i>ADC_raw</i>	Pointer to the raw ADC values array.
<i>measurements</i>	Pointer to the measurements struct to store the results.

**Return values**

<i>OK</i>	0 if an error occurred, 1 if successful.
-----------	--

Here is the call graph for this function:



Here is the caller graph for this function:

**6.49.2.2 getLinear()**

```

float getLinear (
    uint32_t bits,
    float slope,
    float offset )
  
```

Convert ADC reading to physical measurement with linear response.

**Parameters**

<i>bits</i>	The ADC reading.
<i>slope</i>	The slope (units per volt).
<i>offset</i>	The offset (volts at zero).

Return values

<i>measurement</i>	The physical measurement.
--------------------	---------------------------

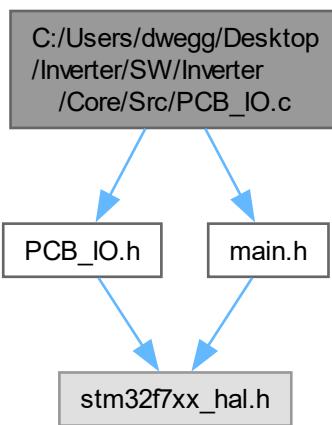
Here is the caller graph for this function:



## 6.50 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/PCB\_IO.c File Reference

This file provides functions for handling GPIOs and other low-priority tasks.

```
#include "PCB_IO.h"
#include "main.h"
Include dependency graph for PCB_IO.c:
```



### Functions

- void `LED_handler (LED *led, uint32_t ms_counter)`  
*LED handler function.*

## Variables

- `LED led_left = { .port = LED_LEFT_GPIO_Port, .pin = LED_LEFT_Pin, .mode = LED_MODE_OFF }`
- `LED led_right = { .port = LED_RIGHT_GPIO_Port, .pin = LED_RIGHT_Pin, .mode = LED_MODE_OFF }`
- `LED led_error = { .port = LED_ERR_GPIO_Port, .pin = LED_ERR_Pin, .mode = LED_MODE_OFF }`

### 6.50.1 Detailed Description

This file provides functions for handling GPIOs and other low-priority tasks.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg in GitHub). All rights reserved.

This software is licensed under the Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. For more information, see: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

### 6.50.2 Function Documentation

#### 6.50.2.1 LED\_handler()

```
void LED_handler (
    LED * led,
    uint32_t ms_counter )
```

`LED` handler function.

This function handles the `LED` blinking modes based on the `LED` mode and current millisecond counter.

#### Parameters

<code>led</code>	Pointer to the <code>LED</code> structure.
<code>ms_counter</code>	Current millisecond counter.

Here is the caller graph for this function:



### 6.50.3 Variable Documentation

#### 6.50.3.1 led\_error

```
LED led_error = { .port = LED_ERR_GPIO_Port, .pin = LED_ERR_Pin, .mode = LED_MODE_OFF }
```

#### 6.50.3.2 led\_left

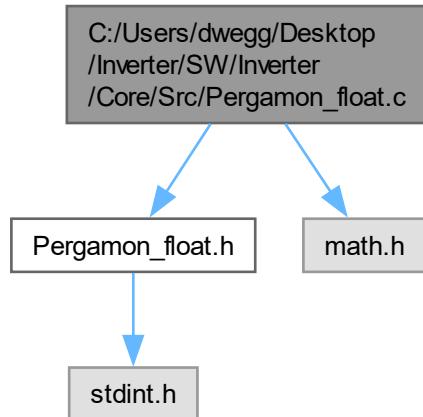
```
LED led_left = { .port = LED_LEFT_GPIO_Port, .pin = LED_LEFT_Pin, .mode = LED_MODE_OFF }
```

#### 6.50.3.3 led\_right

```
LED led_right = { .port = LED_RIGHT_GPIO_Port, .pin = LED_RIGHT_Pin, .mode = LED_MODE_OFF }
```

## 6.51 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/Pergamon\_float.c File Reference

```
#include "Pergamon_float.h"
#include <math.h>
Include dependency graph for Pergamon_float.c:
```



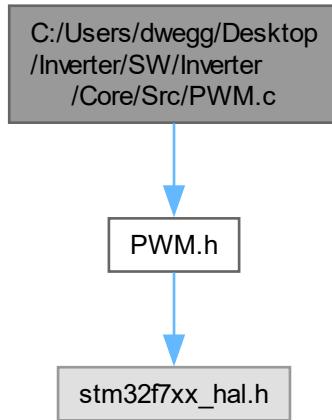
## Functions

- void `pi_aw_calc` (volatile `pi_aw_struct` \*v)  
*Calculates the Proportional-Integral (PI) control with anti-windup.*
- void `pi_init` (volatile `pi_struct` \*v)  
*Initializes the constants of the Proportional-Integral (PI) controller.*
- void `pi_calc` (volatile `pi_struct` \*v)  
*Calculates the Proportional-Integral (PI) control action with feedforward and saturation.*
- void `pi_extsat_calc` (volatile `pi_struct` \*v)  
*Calculates the Proportional-Integral (PI) control action without saturation for external saturation.*
- void `clarke3F_calc` (volatile `clarke3F_struct` \*v)  
*Calculates the Clarke transformation for three-phase signals.*
- void `iclarke3F_calc` (volatile `iclarke3F_struct` \*v)  
*Calculates the inverse Clarke transformation for three-phase signals.*
- void `rot_calc` (volatile `rot_struct` \*v)  
*Calculates the rotation transformation (clockwise).*
- void `irot_calc` (volatile `irot_struct` \*v)  
*Calculates the inverse rotation transformation (counterclockwise).*
- void `angle_calc` (volatile `angle_struct` \*v)  
*Generates the angle.*
- void `svpwm_calc` (volatile `svpwm_struct` \*v)  
*Calculates the Space Vector Pulse Width Modulation (SVPWM).*
- void `rampa_calc` (volatile `rampa_struct` \*v)  
*Calculates the ramp.*
- void `rampa_dual_calc` (volatile `rampa_dual_struct` \*v)  
*Calculates the dual ramp.*
- void `datalog_calc` (volatile `datalog_struct` \*dl)  
*Calculates the data log.*
- void `filtreLP_init` (volatile `filtreLP_struct` \*v)  
*Initializes the first-order filter.*
- void `filtreLP_calc` (volatile `filtreLP_struct` \*v)  
*Calculates the first-order filter.*
- void `avg_calc_10_samples` (volatile `avg_struct_10` \*v)  
*Calculates the average of 10 samples.*
- void `RMS_calc` (volatile `RMS_struct` \*v)  
*Calculates the Root Mean Square (RMS).*
- void `step_calc` (volatile `step_struct` \*v)  
*Calculates the step function.*

## 6.52 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/PWM.c File Reference

This file provides functions for controlling PWM output.

```
#include "PWM.h"  
Include dependency graph for PWM.c:
```



## Functions

- void `enable_PWM` (TIM\_HandleTypeDef \*htim)  
*Enable PWM output.*
- void `disable_PWM` (TIM\_HandleTypeDef \*htim)  
*Disable PWM output.*
- void `Compare3F_calc` (TIM\_HandleTypeDef \*htim, Compare3F\_struct \*v)
- void `set_PWM` (TIM\_HandleTypeDef \*htim, Duties duties)  
*Set PWM duty cycles.*

### 6.52.1 Detailed Description

This file provides functions for controlling PWM output.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg in GitHub). All rights reserved.

This software is licensed under the Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. For more information, see: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

## 6.52.2 Function Documentation

### 6.52.2.1 Compare3F\_calc()

```
void Compare3F_calc (
    TIM_HandleTypeDef * htim,
    Compare3F_struct * v )
```

Here is the caller graph for this function:



### 6.52.2.2 disable\_PWM()

```
void disable_PWM (
    TIM_HandleTypeDef * htim )
```

Disable PWM output.

This function disables PWM output for the specified timer.

#### Parameters

<i>htim</i>	Pointer to the TIM_HandleTypeDef structure.
-------------	---

### 6.52.2.3 enable\_PWM()

```
void enable_PWM (
    TIM_HandleTypeDef * htim )
```

Enable PWM output.

This function enables PWM output for the specified timer.

#### Parameters

<i>htim</i>	Pointer to the TIM_HandleTypeDef structure.
-------------	---

Here is the caller graph for this function:



#### 6.52.2.4 set\_PWM()

```
void set_PWM (   
    TIM_HandleTypeDef * htim,  
    Duties duties )
```

Set PWM duty cycles.

This function sets the duty cycles for the PWM channels.

##### Parameters

<i>htim</i>	Pointer to the TIM_HandleTypeDef structure.
<i>duties</i>	Duties structure containing duty cycle values.

Here is the call graph for this function:



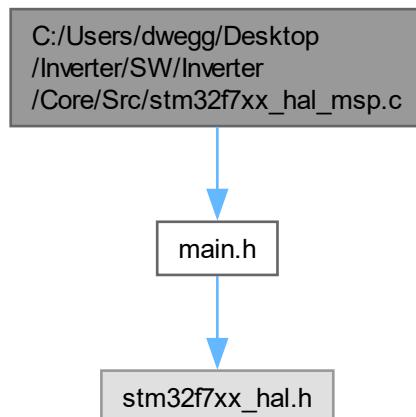
Here is the caller graph for this function:



## 6.53 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/stm32f7xx\_... \_hal\_msp.c File Reference

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

```
#include "main.h"
Include dependency graph for stm32f7xx_hal_msp.c:
```



### Functions

- void [HAL\\_MspInit \(void\)](#)

#### 6.53.1 Detailed Description

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

##### Attention

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

#### 6.53.2 Function Documentation

##### 6.53.2.1 HAL\_MspInit()

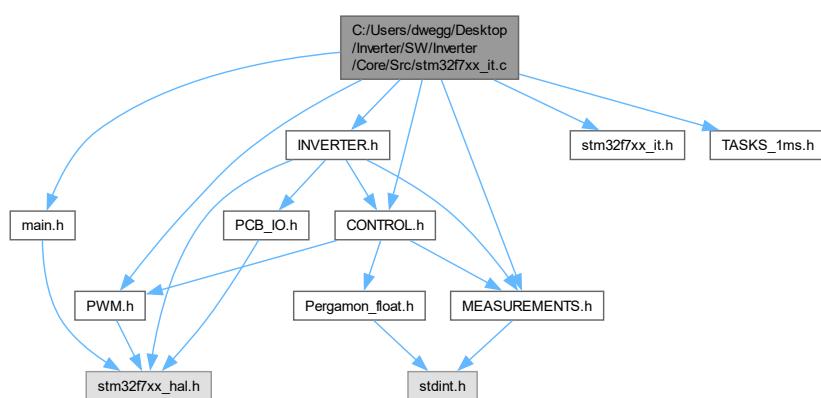
```
void HAL_MspInit (
    void )
```

Initializes the Global MSP.

## 6.54 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/stm32f7xx\_it.c File Reference

Interrupt Service Routines.

```
#include "main.h"
#include "stm32f7xx_it.h"
#include "TASKS_1ms.h"
#include "MEASUREMENTS.h"
#include "PWM.h"
#include "CONTROL.h"
#include "INVERTER.h"
#include "INVERTER.h"
Include dependency graph for stm32f7xx_it.c:
```



### 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 [CAN1\\_RX0\\_IRQHandler](#) (void)

- void [CAN1\\_RX1\\_IRQHandler](#) (void)
 

*This function handles CAN1 RX0 interrupts.*
- void [TIM1\\_UP\\_TIM10\\_IRQHandler](#) (void)
 

*This function handles CAN1 RX1 interrupt.*
- void [TIM6\\_DAC\\_IRQHandler](#) (void)
 

*This function handles TIM1 update interrupt and TIM10 global interrupt.*
- void [DMA2\\_Stream1\\_IRQHandler](#) (void)
 

*This function handles TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts.*
- void [DMA2\\_Stream2\\_IRQHandler](#) (void)
 

*This function handles DMA2 stream1 global interrupt.*
- void [DMA2\\_Stream3\\_IRQHandler](#) (void)
 

*This function handles DMA2 stream2 global interrupt.*

## Variables

- DMA\_HandleTypeDef [hdma\\_adc2](#)
- DMA\_HandleTypeDef [hdma\\_adc3](#)
- CAN\_HandleTypeDef [hcan1](#)
- DAC\_HandleTypeDef [hdac](#)
- TIM\_HandleTypeDef [htim1](#)
- TIM\_HandleTypeDef [htim6](#)

### 6.54.1 Detailed Description

Interrupt Service Routines.

#### Attention

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

### 6.54.2 Function Documentation

#### 6.54.2.1 BusFault\_Handler()

```
void BusFault_Handler (
    void )
```

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

#### 6.54.2.2 CAN1\_RX0\_IRQHandler()

```
void CAN1_RX0_IRQHandler (
    void )
```

This function handles CAN1 RX0 interrupts.

#### 6.54.2.3 CAN1\_RX1\_IRQHandler()

```
void CAN1_RX1_IRQHandler (
    void )
```

This function handles CAN1 RX1 interrupt.

#### 6.54.2.4 DebugMon\_Handler()

```
void DebugMon_Handler (
    void )
```

This function handles Debug monitor.

#### 6.54.2.5 DMA2\_Stream1\_IRQHandler()

```
void DMA2_Stream1_IRQHandler (
    void )
```

This function handles DMA2 stream1 global interrupt.

#### 6.54.2.6 DMA2\_Stream2\_IRQHandler()

```
void DMA2_Stream2_IRQHandler (
    void )
```

This function handles DMA2 stream2 global interrupt.

#### 6.54.2.7 HardFault\_Handler()

```
void HardFault_Handler (
    void )
```

This function handles Hard fault interrupt.

#### 6.54.2.8 MemManage\_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

#### 6.54.2.9 NMI\_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

#### 6.54.2.10 PendSV\_Handler()

```
void PendSV_Handler (
    void )
```

This function handles Pendable request for system service.

#### 6.54.2.11 SVC\_Handler()

```
void SVC_Handler (
    void )
```

This function handles System service call via SWI instruction.

#### 6.54.2.12 SysTick\_Handler()

```
void SysTick_Handler (
    void )
```

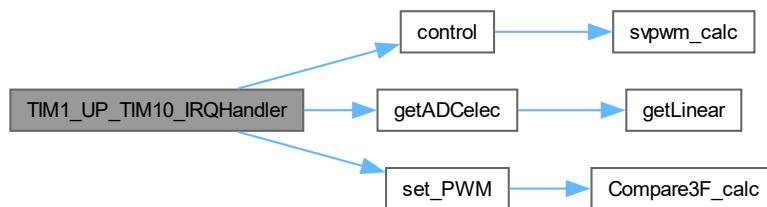
This function handles System tick timer.

#### 6.54.2.13 TIM1\_UP\_TIM10\_IRQHandler()

```
void TIM1_UP_TIM10_IRQHandler (
    void )
```

This function handles TIM1 update interrupt and TIM10 global interrupt.

Here is the call graph for this function:



#### 6.54.2.14 TIM6\_DAC\_IRQHandler()

```
void TIM6_DAC_IRQHandler (
    void )
```

This function handles TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts.

Here is the call graph for this function:



#### 6.54.2.15 UsageFault\_Handler()

```
void UsageFault_Handler (
    void )
```

This function handles Undefined instruction or illegal state.

### 6.54.3 Variable Documentation

#### 6.54.3.1 hcan1

```
CAN_HandleTypeDef hcan1 [extern]
```

#### 6.54.3.2 hdac

```
DAC_HandleTypeDef hdac [extern]
```

#### 6.54.3.3 hdma\_adc2

```
DMA_HandleTypeDef hdma_adc2 [extern]
```

#### 6.54.3.4 hdma\_adc3

```
DMA_HandleTypeDef hdma_adc3 [extern]
```

#### 6.54.3.5 htim1

```
TIM_HandleTypeDef htim1 [extern]
```

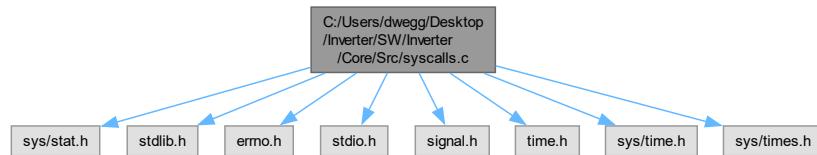
### 6.54.3.6 htim6

```
TIM_HandleTypeDef htim6 [extern]
```

## 6.55 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/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>
Include dependency graph for syscalls.c:
```



## Functions

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

## Variables

- char \*\* **environ** = \_\_env

### 6.55.1 Detailed Description

STM32CubeIDE Minimal System calls file.

#### Author

Auto-generated by STM32CubeIDE

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

#### Attention

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

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

### 6.55.2 Function Documentation

#### 6.55.2.1 \_\_attribute\_\_()

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

Here is the call graph for this function:



### 6.55.2.2 \_\_io\_getchar()

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

Here is the caller graph for this function:



### 6.55.2.3 \_\_io\_putchar()

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

### 6.55.2.4 \_close()

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

### 6.55.2.5 \_execve()

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

### 6.55.2.6 \_exit()

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

Here is the call graph for this function:



**6.55.2.7 \_fork()**

```
int _fork (
    void )
```

**6.55.2.8 \_fstat()**

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

**6.55.2.9 \_getpid()**

```
int _getpid (
    void )
```

**6.55.2.10 \_isatty()**

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

**6.55.2.11 \_kill()**

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

Here is the caller graph for this function:

**6.55.2.12 \_link()**

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

**6.55.2.13 lseek()**

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

**6.55.2.14 open()**

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

**6.55.2.15 stat()**

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

**6.55.2.16 times()**

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

**6.55.2.17 unlink()**

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

**6.55.2.18 wait()**

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

**6.55.2.19 initialise\_monitor\_handles()**

```
void initialise_monitor_handles ( )
```

**6.55.3 Variable Documentation****6.55.3.1 environ**

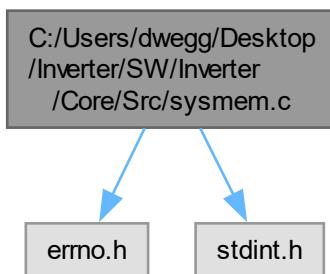
```
char** environ = __env
```

## 6.56 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/sysmem.c File Reference

STM32CubeIDE System Memory calls file.

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

Include dependency graph for sysmem.c:



### Functions

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

#### 6.56.1 Detailed Description

STM32CubeIDE System Memory calls file.

#### Author

Generated by STM32CubeIDE

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

#### Attention

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

## 6.56.2 Function Documentation

### 6.56.2.1 `_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      #
* #      #      #      #      #      #      #      #      #      #      #      #      #
* #      #      #      #      #      #      #      #      #      #      #      #      #
* ##### ^-- 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
-------------------	-------------

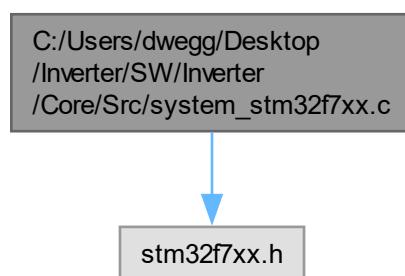
#### Returns

Pointer to allocated memory

## 6.57 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/system\_stm32f7xx.c File Reference

CMSIS Cortex-M7 Device Peripheral Access Layer System Source File.

```
#include "stm32f7xx.h"
Include dependency graph for system_stm32f7xx.c:
```



## Macros

- #define HSE\_VALUE ((uint32\_t)25000000)
- #define HSI\_VALUE ((uint32\_t)16000000)

## Functions

- void [SystemInit](#) (void)  
*Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.*
- void [SystemCoreClockUpdate](#) (void)  
*Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.*

## Variables

- uint32\_t [SystemCoreClock](#) = 16000000
- const uint8\_t [AHBPrescTable](#) [16] = {0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9}
- const uint8\_t [APBPrescTable](#) [8] = {0, 0, 0, 0, 1, 2, 3, 4}

### 6.57.1 Detailed Description

CMSIS Cortex-M7 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\_stm32f7xx.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) 2016 STMicroelectronics. All rights reserved.

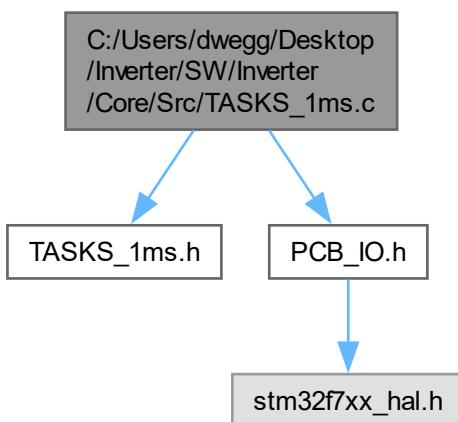
This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

## 6.58 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/TASKS\_1ms.c File Reference

This file contains functions to execute tasks every 1ms.

```
#include "TASKS_1ms.h"
#include "PCB_IO.h"
```

Include dependency graph for TASKS\_1ms.c:



### Functions

- void `tasks_1ms` (void)  
*Function to be executed every 1ms.*

#### 6.58.1 Detailed Description

This file contains functions to execute tasks every 1ms.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg in GitHub). All rights reserved.

This software is licensed under the Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. For more information, see: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

## 6.58.2 Function Documentation

### 6.58.2.1 tasks\_1ms()

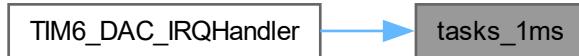
```
void tasks_1ms (
    void )
```

Function to be executed every 1ms.

This function is called by the TIM6 IRQ handler every millisecond. It increments the millisecond counter and calls the [LED](#) handler for left, right, and error LEDs. Here is the call graph for this function:



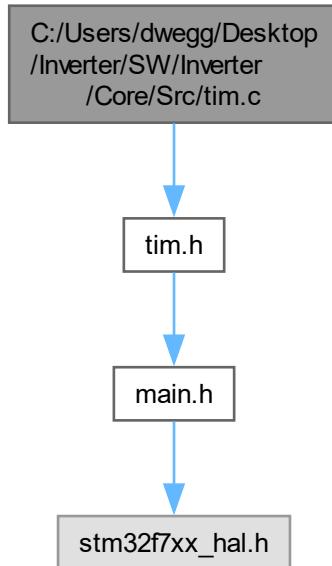
Here is the caller graph for this function:



## 6.59 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/tim.c File Reference

This file provides code for the configuration of the TIM instances.

```
#include "tim.h"
Include dependency graph for tim.c:
```



## Functions

- void [MX\\_TIM1\\_Init](#) (void)
- void [MX\\_TIM2\\_Init](#) (void)
- void [MX\\_TIM4\\_Init](#) (void)
- void [MX\\_TIM6\\_Init](#) (void)
- void [MX\\_TIM8\\_Init](#) (void)
- void [HAL\\_TIM\\_Base\\_MspInit](#) (TIM\_HandleTypeDef \*tim\_baseHandle)
- void [HAL\\_TIM\\_IC\\_MspInit](#) (TIM\_HandleTypeDef \*tim\_icHandle)
- void [HAL\\_TIM\\_MspPostInit](#) (TIM\_HandleTypeDef \*timHandle)
- void [HAL\\_TIM\\_Base\\_MspDeInit](#) (TIM\_HandleTypeDef \*tim\_baseHandle)
- void [HAL\\_TIM\\_IC\\_MspDeInit](#) (TIM\_HandleTypeDef \*tim\_icHandle)

## Variables

- TIM\_HandleTypeDef [htim1](#)
- TIM\_HandleTypeDef [htim2](#)
- TIM\_HandleTypeDef [htim4](#)
- TIM\_HandleTypeDef [htim6](#)
- TIM\_HandleTypeDef [htim8](#)

### 6.59.1 Detailed Description

This file provides code for the configuration of the TIM instances.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

### 6.59.2 Function Documentation

#### 6.59.2.1 HAL\_TIM\_Base\_MspDeInit()

```
void HAL_TIM_Base_MspDeInit (
    TIM_HandleTypeDef * tim_baseHandle )
```

Uncomment the line below to disable the "TIM6\_DAC\_IRQHandler" interrupt Be aware, disabling shared interrupt may affect other IPs

#### 6.59.2.2 HAL\_TIM\_Base\_MspInit()

```
void HAL_TIM_Base_MspInit (
    TIM_HandleTypeDef * tim_baseHandle )
```

#### 6.59.2.3 HAL\_TIM\_IC\_MspDeInit()

```
void HAL_TIM_IC_MspDeInit (
    TIM_HandleTypeDef * tim_icHandle )
```

TIM2 GPIO Configuration PB10 -----> TIM2\_CH3 PA15 -----> TIM2\_CH1

TIM4 GPIO Configuration PD12 -----> TIM4\_CH1 PD14 -----> TIM4\_CH3

#### 6.59.2.4 HAL\_TIM\_IC\_MspInit()

```
void HAL_TIM_IC_MspInit (
    TIM_HandleTypeDef * tim_icHandle )
```

TIM2 GPIO Configuration PB10 -----> TIM2\_CH3 PA15 -----> TIM2\_CH1

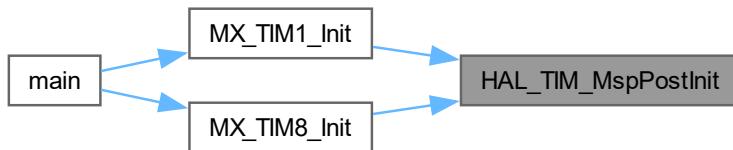
TIM4 GPIO Configuration PD12 -----> TIM4\_CH1 PD14 -----> TIM4\_CH3

### 6.59.2.5 HAL\_TIM\_MspPostInit()

```
void HAL_TIM_MspPostInit (
    TIM_HandleTypeDef * timHandle )
```

TIM1 GPIO Configuration PE8 -----> TIM1\_CH1N PE9 -----> TIM1\_CH1 PE10 -----> TIM1\_CH2N PE11 ----->  
TIM1\_CH2 PE12 -----> TIM1\_CH3N PE13 -----> TIM1\_CH3

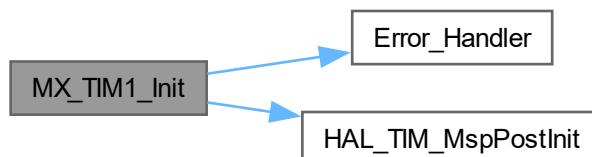
TIM8 GPIO Configuration PA5 -----> TIM8\_CH1N PB14 -----> TIM8\_CH2N PB15 -----> TIM8\_CH3N PC6 ----->  
TIM8\_CH1 PC7 -----> TIM8\_CH2 PC8 -----> TIM8\_CH3Here is the caller graph for this function:



### 6.59.2.6 MX\_TIM1\_Init()

```
void MX_TIM1_Init (
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.59.2.7 MX\_TIM2\_Init()

```
void MX_TIM2_Init (
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.59.2.8 MX\_TIM4\_Init()

```
void MX_TIM4_Init (
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.59.2.9 MX\_TIM6\_Init()

```
void MX_TIM6_Init (
    void )
```

Here is the call graph for this function:



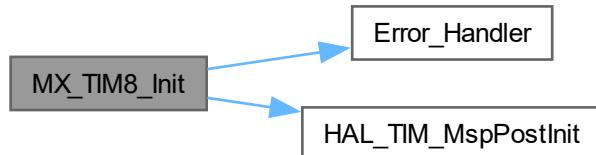
Here is the caller graph for this function:



### 6.59.2.10 MX\_TIM8\_Init()

```
void MX_TIM8_Init (
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.59.3 Variable Documentation

#### 6.59.3.1 htim1

```
TIM_HandleTypeDef htim1
```

#### 6.59.3.2 htim2

```
TIM_HandleTypeDef htim2
```

#### 6.59.3.3 htim4

```
TIM_HandleTypeDef htim4
```

#### 6.59.3.4 htim6

```
TIM_HandleTypeDef htim6
```

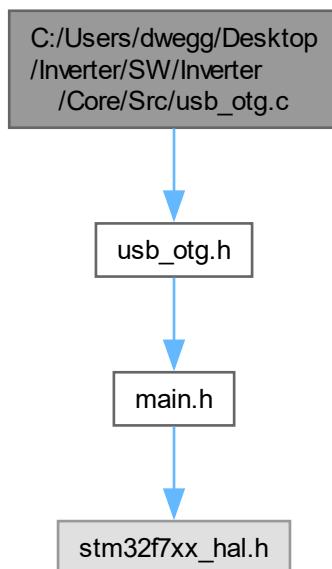
#### 6.59.3.5 htim8

```
TIM_HandleTypeDef htim8
```

## 6.60 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/usb\_otg.c File Reference

This file provides code for the configuration of the USB\_OTG instances.

```
#include "usb_otg.h"  
Include dependency graph for usb_otg.c:
```



### Functions

- void [MX\\_USB\\_OTG\\_FS\\_USB\\_Init](#) (void)

#### 6.60.1 Detailed Description

This file provides code for the configuration of the USB\_OTG instances.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

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

## 6.60.2 Function Documentation

### 6.60.2.1 MX\_USB\_OTG\_FS\_USB\_Init()

```
void MX_USB_OTG_FS_USB_Init (
    void )
```

Here is the caller graph for this function:





# Index

- Integral Controllers, 9
  - pi\_aw\_calc, 9
  - pi\_calc, 10
  - pi\_extsat\_calc, 10
  - pi\_init, 10
- \_attribute\_
  - syscalls.c, 195
- \_io\_getchar
  - syscalls.c, 195
- \_io\_putchar
  - syscalls.c, 196
- \_close
  - syscalls.c, 196
- \_execve
  - syscalls.c, 196
- \_exit
  - syscalls.c, 196
- \_fork
  - syscalls.c, 196
- \_fstat
  - syscalls.c, 197
- \_getpid
  - syscalls.c, 197
- \_isatty
  - syscalls.c, 197
- \_kill
  - syscalls.c, 197
- \_link
  - syscalls.c, 197
- \_lseek
  - syscalls.c, 197
- \_open
  - syscalls.c, 198
- \_sbrk
  - sysmem.c, 200
- \_stat
  - syscalls.c, 198
- \_times
  - syscalls.c, 198
- \_unlink
  - syscalls.c, 198
- \_wait
  - syscalls.c, 198

- A
- Encoder, 34
- a
- clarke3F\_struct, 29
- iclarke3F\_struct, 37
- A\_L\_GPIO\_Port
- main.h, 89
- A\_L\_Pin
- main.h, 89
- A\_R\_GPIO\_Port
- main.h, 89
- A\_R\_Pin
- main.h, 89
- adc.c
- hadc1, 157
- hadc2, 157
- hadc3, 158
- HAL\_ADC\_MspDeInit, 155
- HAL\_ADC\_MspInit, 155
- hdma\_adc2, 158
- hdma\_adc3, 158
- MX\_ADC1\_Init, 155
- MX\_ADC2\_Init, 156
- MX\_ADC3\_Init, 157
- adc.h
- hadc1, 63
- hadc2, 63
- hadc3, 63
- MX\_ADC1\_Init, 61
- MX\_ADC2\_Init, 61
- MX\_ADC3\_Init, 62
- ADC\_raw
- inverterStruct, 38
- AHBPrescTable
- STM32F7xx\_System\_Private\_Variables, 24
- alfa
- filtreLP\_struct, 35
- alfaA
- Compare3F\_struct, 30
- alfaB
- Compare3F\_struct, 30
- alfaC
- Compare3F\_struct, 30
- Angle
- RMS\_struct, 52
- angle
- angle\_struct, 27
- Angle\_ant
- RMS\_struct, 52
- angle\_calc
- Utility Functions, 15
- angle\_struct
- angle, 27
- calc, 27
- freq, 27

Ts, 28

APBPrescTable  
  STM32F7xx\_System\_Private\_Variables, 24

ART\_ACCELERATOR\_ENABLE  
  stm32f7xx\_hal\_conf.h, 121

assert\_param  
  stm32f7xx\_hal\_conf.h, 121

avg\_calc\_10\_samples  
  Signal Processing Functions, 17

avg\_struct\_10, 28  
  in, 28  
  out, 28

B  
  Encoder, 34

b  
  clarke3F\_struct, 29  
  iclarke3F\_struct, 37

B\_L\_GPIO\_Port  
  main.h, 89

B\_L\_Pin  
  main.h, 89

B\_R\_GPIO\_Port  
  main.h, 89

B\_R\_Pin  
  main.h, 89

BusFault\_Handler  
  stm32f7xx\_it.c, 190  
  stm32f7xx\_it.h, 139

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/adc.h, 59, 63

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/can.h, 64, 66

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/CONTROL.h, 66, 69

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/dac.h, 69, 71

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/dma.h, 72, 74

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/FSM.h, 74, 76

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/gpio.h, 77, 79

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/i2c.h, 79, 81

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/INVERTER.h, 82, 85

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/main.h, 86, 98

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/MEASUREMENTS.h, 100, 104

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PCB\_IO.h, 104, 108

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon\_float.h, 109, 111

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PWM.h, 114, 118

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/stm32f7xx\_hal\_co  
  119, 133

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/stm32f7xx\_it.h,  
  138, 142

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/TASKS\_1ms.h,  
  143, 144

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/tim.h,  
  145, 150

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/usb\_otg.h,  
  151, 153

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/adc.c,  
  153

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/can.c,  
  158

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/CONTROL.c,  
  160

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/dac.c,  
  162

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/dma.c,  
  165

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/FSM.c,  
  166

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/gpio.c,  
  168

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/i2c.c,  
  169

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/INVERTER.c,  
  171

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/main.c,  
  174

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/MEASUREMENT.  
  178

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/PCB\_IO.c,  
  181

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/Pergamon\_float.c,  
  183

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/PWM.c,  
  184

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/stm32f7xx\_hal\_m  
  188

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/stm32f7xx\_it.c,  
  189

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/syscalls.c,  
  194

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/sysmem.c,  
  199

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/system\_stm32f7x  
  200

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/TASKS\_1ms.c,  
  202

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/tim.c,  
  203

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/usb\_otg.c,  
  210

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/usb\_otg.h,  
  210

angle\_struct, 27

clarke3F\_struct, 29

Compare3F\_struct, 30

datalog\_struct, 31  
filtreLP\_struct, 35  
iclarke3F\_struct, 37  
irot\_struct, 40  
pi\_aw\_struct, 44  
pi\_struct, 47  
rampa\_dual\_struct, 49  
rampa\_struct, 51  
rot\_struct, 53  
step\_struct, 55  
svpwm\_struct, 57

can.c  
    HAL\_CAN\_MspDeInit, 159  
    HAL\_CAN\_MspInit, 159  
    hcan1, 160  
    MX\_CAN1\_Init, 159

can.h  
    hcan1, 66  
    MX\_CAN1\_Init, 65

CAN1\_RX0\_IRQHandler  
    stm32f7xx\_it.c, 190  
    stm32f7xx\_it.h, 139

CAN1\_RX1\_IRQHandler  
    stm32f7xx\_it.c, 190  
    stm32f7xx\_it.h, 139

Clarke and Park Transformations, 11  
    clarke3F\_calc, 12  
    iclarke3F\_calc, 13  
    irot\_calc, 13  
    rot\_calc, 14

clarke3F\_calc  
    Clarke and Park Transformations, 12

clarke3F\_struct, 29  
    a, 29  
    b, 29  
    calc, 29  
    D, 29  
    Q, 29

CMSIS, 21

compA  
    Compare3F\_struct, 30

Compare3F\_calc  
    PWM.c, 186  
    PWM.h, 116

Compare3F\_struct, 30  
    alfaA, 30  
    alfaB, 30  
    alfaC, 30  
    calc, 30  
    compA, 30  
    compB, 31  
    compC, 31

compares\_LEFT  
    PWM.h, 118

compares\_RIGHT  
    PWM.h, 118

compB  
    Compare3F\_struct, 31

compC  
    Compare3F\_struct, 31

control  
    CONTROL.c, 161  
    CONTROL.h, 68

CONTROL.c  
    control, 161  
    valfa\_L, 162  
    vbeta\_L, 162

CONTROL.h  
    control, 68  
    valfa\_L, 69  
    vbeta\_L, 69

cosFi  
    irot\_struct, 40  
    rot\_struct, 53

Counter  
    step\_struct, 55

CURRENT\_OFFSET  
    MEASUREMENTS.h, 102

CURRENT\_SLOPE  
    MEASUREMENTS.h, 102

D

Da  
    Duties, 33

dac.c  
    HAL\_DAC\_MspDeInit, 164  
    HAL\_DAC\_MspInit, 164  
    hdac, 165  
    MX\_DAC\_Init, 164

dac.h  
    hdac, 71  
    MX\_DAC\_Init, 71

DAC\_GPIO\_Port  
    main.h, 89

DAC\_Pin  
    main.h, 89

datalog\_calc  
    Signal Processing Functions, 17

datalog\_struct, 31  
    calc, 31  
    estat, 31  
    i, 31  
    j, 32  
    log, 32  
    prescaler, 32  
    var, 32

Db  
    Duties, 33

Dc  
    Duties, 33

DebugMon\_Handler  
     stm32f7xx\_it.c, 191  
     stm32f7xx\_it.h, 140  
 Decr  
     rampa\_dual\_struct, 49  
 DIR  
     Encoder, 34  
 DIR\_GPIO\_Port  
     main.h, 89  
 DIR\_Pin  
     main.h, 90  
 DIR\_STATE  
     PCB\_IO.h, 106  
 DISABLE  
     PCB\_IO.h, 106  
 disable\_PWM  
     PWM.c, 186  
     PWM.h, 116  
 DIV2  
     Math Constants, 7  
 dma.c  
     MX\_DMA\_Init, 166  
 dma.h  
     MX\_DMA\_Init, 73  
 DMA2\_Stream1\_IRQHandler  
     stm32f7xx\_it.c, 191  
     stm32f7xx\_it.h, 140  
 DMA2\_Stream2\_IRQHandler  
     stm32f7xx\_it.c, 191  
     stm32f7xx\_it.h, 140  
 DP83848\_PHY\_ADDRESS  
     stm32f7xx\_hal\_conf.h, 122  
 DT  
     main.h, 90  
 Duties, 32  
     Da, 33  
     Db, 33  
     Dc, 33  
 duties  
     inverterStruct, 38  
 e  
     pi\_aw\_struct, 44  
     pi\_struct, 47  
 ENABLE  
     PCB\_IO.h, 106  
 enable  
     filtreLP\_struct, 35  
     pi\_aw\_struct, 44  
     pi\_struct, 47  
     rampa\_dual\_struct, 49  
     rampa\_struct, 51  
     step\_struct, 55  
 ENABLE\_L\_GPIO\_Port  
     main.h, 90  
 ENABLE\_L\_Pin  
     main.h, 90  
 enable\_pin  
     inverterStruct, 39  
     enable\_port  
         inverterStruct, 39  
 enable\_PWM  
     PWM.c, 186  
     PWM.h, 117  
 ENABLE\_R\_GPIO\_Port  
     main.h, 90  
 ENABLE\_R\_Pin  
     main.h, 90  
 Encoder, 33  
     A, 34  
     B, 34  
     DIR, 34  
     theta\_e, 34  
     theta\_m, 34  
     we, 34  
     wm\_rpm, 34  
     Z, 34  
 encoder  
     inverterStruct, 39  
 environ  
     syscalls.c, 198  
 Error\_Handler  
     main.c, 175  
     main.h, 97  
 estat  
     datalog\_struct, 31  
 ETH\_RX\_BUF\_SIZE  
     stm32f7xx\_hal\_conf.h, 122  
 ETH\_RXBUFN  
     stm32f7xx\_hal\_conf.h, 122  
 ETH\_TX\_BUF\_SIZE  
     stm32f7xx\_hal\_conf.h, 122  
 ETH\_TXBUFN  
     stm32f7xx\_hal\_conf.h, 122  
 EXTERNAL\_CLOCK\_VALUE  
     stm32f7xx\_hal\_conf.h, 122  
 fc  
     filtreLP\_struct, 36  
 filtrep\_struct  
     Signal Processing Functions, 18  
 filtrep\_init  
     Signal Processing Functions, 18  
 filtrep\_struct, 35  
     alfa, 35  
     calc, 35  
     enable, 35  
     fc, 36  
     in, 36  
     init, 36  
     out, 36  
     Ts, 36  
 Freq  
     RMS\_struct, 52  
 freq  
     angle\_struct, 27  
 fs  
     step\_struct, 55

FSM.c  
    inv\_FSM, 167  
FSM.h  
    inv\_FSM, 76  
  
getADCelec  
    MEASUREMENTS.c, 179  
    MEASUREMENTS.h, 102  
getLinear  
    MEASUREMENTS.c, 180  
    MEASUREMENTS.h, 103  
gpio.c  
    MX\_GPIO\_Init, 169  
gpio.h  
    MX\_GPIO\_Init, 78  
  
hadc  
    inverterStruct, 39  
hadc1  
    adc.c, 157  
    adc.h, 63  
hadc2  
    adc.c, 157  
    adc.h, 63  
hadc3  
    adc.c, 158  
    adc.h, 63  
HAL\_ADC\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 122  
HAL\_ADC\_MspDelInit  
    adc.c, 155  
HAL\_ADC\_MspInit  
    adc.c, 155  
HAL\_CAN\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 122  
HAL\_CAN\_MspDelInit  
    can.c, 159  
HAL\_CAN\_MspInit  
    can.c, 159  
HAL\_CORTEX\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 122  
HAL\_DAC\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 123  
HAL\_DAC\_MspDelInit  
    dac.c, 164  
HAL\_DAC\_MspInit  
    dac.c, 164  
HAL\_DMA\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 123  
HAL\_EXTI\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 123  
HAL\_FLASH\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 123  
HAL\_GPIO\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 123  
HAL\_I2C\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 123  
HAL\_I2C\_MspDelInit  
    i2c.c, 170  
HAL\_I2C\_MspInit  
    i2c.c, 170  
HAL\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 123  
HAL\_MspInit  
    stm32f7xx\_hal\_msp.c, 188  
HAL\_PWR\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 123  
HAL\_RCC\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 123  
HAL\_TIM\_Base\_MspDelInit  
    tim.c, 205  
HAL\_TIM\_Base\_MspInit  
    tim.c, 205  
HAL\_TIM\_IC\_MspDelInit  
    tim.c, 205  
HAL\_TIM\_IC\_MspInit  
    tim.c, 205  
HAL\_TIM\_MODULE\_ENABLED  
    stm32f7xx\_hal\_conf.h, 123  
HAL\_TIM\_MspPostInit  
    tim.c, 205  
    tim.h, 146  
HardFault\_Handler  
    stm32f7xx\_it.c, 191  
    stm32f7xx\_it.h, 140  
hcan1  
    can.c, 160  
    can.h, 66  
    stm32f7xx\_it.c, 193  
hdac  
    dac.c, 165  
    dac.h, 71  
    stm32f7xx\_it.c, 193  
hdma\_adc2  
    adc.c, 158  
    stm32f7xx\_it.c, 193  
hdma\_adc3  
    adc.c, 158  
    stm32f7xx\_it.c, 193  
hi2c1  
    i2c.c, 171  
    i2c.h, 81  
HSE\_STARTUP\_TIMEOUT  
    stm32f7xx\_hal\_conf.h, 124  
HSE\_VALUE  
    stm32f7xx\_hal\_conf.h, 124  
    STM32F7xx\_System\_Private\_Includes, 22  
HSI\_VALUE  
    stm32f7xx\_hal\_conf.h, 124  
    STM32F7xx\_System\_Private\_Includes, 22  
htim  
    inverterStruct, 39  
htim1  
    stm32f7xx\_it.c, 193  
    tim.c, 209  
    tim.h, 150  
htim2

tim.c, 209  
 tim.h, 150  
 htim4  
     tim.c, 209  
     tim.h, 150  
 htim6  
     stm32f7xx\_it.c, 193  
     tim.c, 209  
     tim.h, 150  
 htim8  
     tim.c, 209  
     tim.h, 150  
 i  
     datalog\_struct, 31  
 i2c.c  
     HAL\_I2C\_MspDeInit, 170  
     HAL\_I2C\_MspInit, 170  
     hi2c1, 171  
     MX\_I2C1\_Init, 170  
 i2c.h  
     hi2c1, 81  
     MX\_I2C1\_Init, 81  
 ia  
     Measurements, 42  
 ia\_L\_GPIO\_Port  
     main.h, 90  
 ia\_L\_Pin  
     main.h, 90  
 ia\_R\_GPIO\_Port  
     main.h, 90  
 ia\_R\_Pin  
     main.h, 90  
 ib  
     Measurements, 42  
 ib\_L\_GPIO\_Port  
     main.h, 91  
 ib\_L\_Pin  
     main.h, 91  
 ib\_R\_GPIO\_Port  
     main.h, 91  
 ib\_R\_Pin  
     main.h, 91  
 ic  
     Measurements, 43  
 ic\_L\_GPIO\_Port  
     main.h, 91  
 ic\_L\_Pin  
     main.h, 91  
 ic\_R\_GPIO\_Port  
     main.h, 91  
 ic\_R\_Pin  
     main.h, 91  
 iclarke3F\_calc  
     Clarke and Park Transformations, 13  
 iclarke3F\_struct, 36  
     a, 37  
     b, 37  
     calc, 37  
 In  
     D, 37  
     Q, 37  
 in  
     step\_struct, 55  
 in  
     avg\_struct\_10, 28  
     filtreLP\_struct, 36  
     rampa\_dual\_struct, 50  
     rampa\_struct, 51  
 Incr  
     rampa\_dual\_struct, 50  
     rampa\_struct, 51  
 init  
     filtreLP\_struct, 36  
     pi\_struct, 47  
 initialise\_monitor\_handles  
     syscalls.c, 198  
 INV3  
     Math Constants, 7  
 INV\_DEG  
     Math Constants, 7  
 inv\_FSM  
     FSM.c, 167  
     FSM.h, 76  
 inv\_init  
     INVERTER.c, 173  
     INVERTER.h, 84  
 INV\_STATEFAULT  
     INVERTER.h, 84  
 INV\_STATE\_IDLE  
     INVERTER.h, 84  
 INV\_STATE\_RUNNING  
     INVERTER.h, 84  
 INV\_STATE\_STARTUP  
     INVERTER.h, 84  
 INVERTER.c  
     inv\_init, 173  
     invLeft, 173  
     invRight, 173  
 INVERTER.h  
     inv\_init, 84  
     INV\_STATE\_FAULT, 84  
     INV\_STATE\_IDLE, 84  
     INV\_STATE\_RUNNING, 84  
     INV\_STATE\_STARTUP, 84  
     inverterState, 84  
     invLeft, 85  
     invRight, 85  
 inverterState  
     INVERTER.h, 84  
 inverterStruct, 38  
     ADC\_raw, 38  
     duties, 38  
     enable\_pin, 39  
     enable\_port, 39  
     encoder, 39  
     hadc, 39  
     htim, 39

led, 39  
measurements, 39  
state, 39  
invLeft  
    INVERTER.c, 173  
    INVERTER.h, 85  
invRight  
    INVERTER.c, 173  
    INVERTER.h, 85  
IPI  
    Math Constants, 8  
IPI2  
    Math Constants, 8  
irot\_calc  
    Clarke and Park Transformations, 13  
irot\_struct, 40  
    calc, 40  
    cosFi, 40  
    D, 41  
    d, 40  
    Q, 41  
    q, 41  
    sinFi, 41  
ISQ2  
    Math Constants, 8  
ISQ3  
    Math Constants, 8  
j  
    datalog\_struct, 32  
K0  
    pi\_struct, 47  
K1  
    pi\_struct, 47  
Kaw  
    pi\_aw\_struct, 44  
Ki  
    pi\_aw\_struct, 44  
    pi\_struct, 47  
Kp  
    pi\_aw\_struct, 44  
    pi\_struct, 47  
LED, 41  
    mode, 42  
    pin, 42  
    port, 42  
led  
    inverterStruct, 39  
LED\_ERR\_GPIO\_Port  
    main.h, 91  
LED\_ERR\_Pin  
    main.h, 91  
led\_error  
    PCB\_IO.c, 183  
    PCB\_IO.h, 108  
LED\_handler  
    PCB\_IO.c, 182  
    PCB\_IO.h, 107  
led\_left  
    PCB\_IO.c, 183  
    PCB\_IO.h, 108  
LED\_LEFT\_GPIO\_Port  
    main.h, 92  
LED\_LEFT\_Pin  
    main.h, 92  
LED\_MODE\_BLINK\_FAST  
    PCB\_IO.h, 107  
LED\_MODE\_BLINK\_SLOW  
    PCB\_IO.h, 107  
LED\_MODE\_OFF  
    PCB\_IO.h, 107  
LED\_MODE\_ON  
    PCB\_IO.h, 107  
led\_right  
    PCB\_IO.c, 183  
    PCB\_IO.h, 108  
LED\_RIGHT\_GPIO\_Port  
    main.h, 92  
LED\_RIGHT\_Pin  
    main.h, 92  
LEDMode  
    PCB\_IO.h, 107  
log  
    datalog\_struct, 32  
LSE\_STARTUP\_TIMEOUT  
    stm32f7xx\_hal\_conf.h, 124  
LSE\_VALUE  
    stm32f7xx\_hal\_conf.h, 124  
LSI\_VALUE  
    stm32f7xx\_hal\_conf.h, 124  
MAC\_ADDR0  
    stm32f7xx\_hal\_conf.h, 125  
MAC\_ADDR1  
    stm32f7xx\_hal\_conf.h, 125  
MAC\_ADDR2  
    stm32f7xx\_hal\_conf.h, 125  
MAC\_ADDR3  
    stm32f7xx\_hal\_conf.h, 125  
MAC\_ADDR4  
    stm32f7xx\_hal\_conf.h, 125  
MAC\_ADDR5  
    stm32f7xx\_hal\_conf.h, 125  
main  
    main.c, 176  
main.c  
    Error\_Handler, 175  
    main, 176  
    SystemClock\_Config, 177  
main.h  
    A\_L\_GPIO\_Port, 89  
    A\_L\_Pin, 89  
    A\_R\_GPIO\_Port, 89  
    A\_R\_Pin, 89  
    B\_L\_GPIO\_Port, 89  
    B\_L\_Pin, 89

B\_R\_GPIO\_Port, 89  
 B\_R\_Pin, 89  
 DAC\_GPIO\_Port, 89  
 DAC\_Pin, 89  
 DIR\_GPIO\_Port, 89  
 DIR\_Pin, 90  
 DT, 90  
 ENABLE\_L\_GPIO\_Port, 90  
 ENABLE\_L\_Pin, 90  
 ENABLE\_R\_GPIO\_Port, 90  
 ENABLE\_R\_Pin, 90  
 Error\_Handler, 97  
 ia\_L\_GPIO\_Port, 90  
 ia\_L\_Pin, 90  
 ia\_R\_GPIO\_Port, 90  
 ia\_R\_Pin, 90  
 ib\_L\_GPIO\_Port, 91  
 ib\_L\_Pin, 91  
 ib\_R\_GPIO\_Port, 91  
 ib\_R\_Pin, 91  
 ic\_L\_GPIO\_Port, 91  
 ic\_L\_Pin, 91  
 ic\_R\_GPIO\_Port, 91  
 ic\_R\_Pin, 91  
 LED\_ERR\_GPIO\_Port, 91  
 LED\_ERR\_Pin, 91  
 LED\_LEFT\_GPIO\_Port, 92  
 LED\_LEFT\_Pin, 92  
 LED\_RIGHT\_GPIO\_Port, 92  
 LED\_RIGHT\_Pin, 92  
 PWM1\_L\_GPIO\_Port, 92  
 PWM1\_L\_Pin, 92  
 PWM1\_R\_GPIO\_Port, 92  
 PWM1\_R\_Pin, 92  
 PWM2\_L\_GPIO\_Port, 92  
 PWM2\_L\_Pin, 92  
 PWM2\_R\_GPIO\_Port, 93  
 PWM2\_R\_Pin, 93  
 PWM3\_L\_GPIO\_Port, 93  
 PWM3\_L\_Pin, 93  
 PWM3\_R\_GPIO\_Port, 93  
 PWM3\_R\_Pin, 93  
 PWM4\_L\_GPIO\_Port, 93  
 PWM4\_L\_Pin, 93  
 PWM4\_R\_GPIO\_Port, 93  
 PWM4\_R\_Pin, 93  
 PWM5\_L\_GPIO\_Port, 94  
 PWM5\_L\_Pin, 94  
 PWM5\_R\_GPIO\_Port, 94  
 PWM5\_R\_Pin, 94  
 PWM6\_L\_GPIO\_Port, 94  
 PWM6\_L\_Pin, 94  
 PWM6\_R\_GPIO\_Port, 94  
 PWM6\_R\_Pin, 94  
 SC\_det\_GPIO\_Port, 94  
 SC\_det\_Pin, 94  
 Tinv\_L\_GPIO\_Port, 95  
 Tinv\_L\_Pin, 95  
 Tinv\_R\_GPIO\_Port, 95  
 Tinv\_R\_Pin, 95  
 Tmot\_L\_GPIO\_Port, 95  
 Tmot\_L\_Pin, 95  
 Tmot\_R\_GPIO\_Port, 95  
 Tmot\_R\_Pin, 95  
 TRIP\_L\_GPIO\_Port, 95  
 TRIP\_L\_Pin, 95  
 TRIP\_R\_GPIO\_Port, 96  
 TRIP\_R\_Pin, 96  
 TS, 96  
 VDC\_L\_GPIO\_Port, 96  
 VDC\_L\_Pin, 96  
 VDC\_R\_GPIO\_Port, 96  
 VDC\_R\_Pin, 96  
 WRN\_L\_GPIO\_Port, 96  
 WRN\_L\_Pin, 96  
 WRN\_R\_GPIO\_Port, 96  
 WRN\_R\_Pin, 97  
 Z\_L\_GPIO\_Port, 97  
 Z\_L\_Pin, 97  
 Z\_R\_GPIO\_Port, 97  
 Z\_R\_Pin, 97  
 Math Constants, 7  
 DIV2, 7  
 INV3, 7  
 INV\_DEG, 7  
 IPI, 8  
 IPI2, 8  
 ISQ2, 8  
 ISQ3, 8  
 PI, 8  
 PI2, 8  
 SQ2, 8  
 SQ3, 8  
 Measure  
 RMS\_struct, 52  
 Measurements, 42  
 ia, 42  
 ib, 42  
 ic, 43  
 VDC, 43  
 measurements  
 inverterStruct, 39  
 MEASUREMENTS.c  
 getADCelec, 179  
 getLinear, 180  
 MEASUREMENTS.h  
 CURRENT\_OFFSET, 102  
 CURRENT\_SLOPE, 102  
 getADCelec, 102  
 getLinear, 103  
 VOLTAGE\_OFFSET, 102  
 VOLTAGE\_SLOPE, 102  
 MemManage\_Handler  
 stm32f7xx\_it.c, 191  
 stm32f7xx\_it.h, 140  
 Miscellaneous Functions, 20

step\_calc, 20  
mode  
    LED, 42  
MX\_ADC1\_Init  
    adc.c, 155  
    adc.h, 61  
MX\_ADC2\_Init  
    adc.c, 156  
    adc.h, 61  
MX\_ADC3\_Init  
    adc.c, 157  
    adc.h, 62  
MX\_CAN1\_Init  
    can.c, 159  
    can.h, 65  
MX\_DAC\_Init  
    dac.c, 164  
    dac.h, 71  
MX\_DMA\_Init  
    dma.c, 166  
    dma.h, 73  
MX\_GPIO\_Init  
    gpio.c, 169  
    gpio.h, 78  
MX\_I2C1\_Init  
    i2c.c, 170  
    i2c.h, 81  
MX\_TIM1\_Init  
    tim.c, 206  
    tim.h, 146  
MX\_TIM2\_Init  
    tim.c, 206  
    tim.h, 147  
MX\_TIM4\_Init  
    tim.c, 207  
    tim.h, 148  
MX\_TIM6\_Init  
    tim.c, 208  
    tim.h, 148  
MX\_TIM8\_Init  
    tim.c, 208  
    tim.h, 149  
MX\_USB\_OTG\_FS\_USB\_Init  
    usb\_otg.c, 211  
    usb\_otg.h, 152

N\_DATALOG  
    Signal Processing Functions, 17  
NMI\_Handler  
    stm32f7xx\_it.c, 191  
    stm32f7xx\_it.h, 140

Out  
    step\_struct, 55  
out  
    avg\_struct\_10, 28  
    filtreLP\_struct, 36  
    rampa\_dual\_struct, 50  
    rampa\_struct, 51

Out\_RMS  
    RMS\_struct, 52

PCB\_IO.c  
    led\_error, 183  
    LED\_handler, 182  
    led\_left, 183  
    led\_right, 183

PCB\_IO.h  
    DIR\_STATE, 106  
    DISABLE, 106  
    ENABLE, 106  
    led\_error, 108  
    LED\_handler, 107  
    led\_left, 108  
    LED\_MODE\_BLINK\_FAST, 107  
    LED\_MODE\_BLINK\_SLOW, 107  
    LED\_MODE\_OFF, 107  
    LED\_MODE\_ON, 107  
    led\_right, 108  
    LEDMode, 107  
    SC\_DET\_STATE, 106  
    WRN\_STATE, 107

PendSV\_Handler  
    stm32f7xx\_it.c, 191  
    stm32f7xx\_it.h, 140

PHY\_AUTONEGO\_COMPLETE  
    stm32f7xx\_hal\_conf.h, 125

PHY\_AUTONEGOTIATION  
    stm32f7xx\_hal\_conf.h, 125

PHY\_BCR  
    stm32f7xx\_hal\_conf.h, 126

PHY\_BSR  
    stm32f7xx\_hal\_conf.h, 126

PHY\_CONFIG\_DELAY  
    stm32f7xx\_hal\_conf.h, 126

PHY\_DUPLEX\_STATUS  
    stm32f7xx\_hal\_conf.h, 126

PHY\_FULLDUPLEX\_100M  
    stm32f7xx\_hal\_conf.h, 126

PHY\_FULLDUPLEX\_10M  
    stm32f7xx\_hal\_conf.h, 126

PHY\_HALFDUPLEX\_100M  
    stm32f7xx\_hal\_conf.h, 126

PHY\_HALFDUPLEX\_10M  
    stm32f7xx\_hal\_conf.h, 127

PHY\_ISOLATE  
    stm32f7xx\_hal\_conf.h, 127

PHY\_JABBER\_DETECTION  
    stm32f7xx\_hal\_conf.h, 127

PHY\_LINKED\_STATUS  
    stm32f7xx\_hal\_conf.h, 127

PHY\_LOOPBACK  
    stm32f7xx\_hal\_conf.h, 127

PHY\_POWERDOWN  
    stm32f7xx\_hal\_conf.h, 127

PHY\_READ\_TO  
    stm32f7xx\_hal\_conf.h, 127

PHY\_RESET

stm32f7xx\_hal\_conf.h, 128  
PHY\_RESET\_DELAY  
    stm32f7xx\_hal\_conf.h, 128  
PHY\_RESTART\_AUTONEGOTIATION  
    stm32f7xx\_hal\_conf.h, 128  
PHY\_SPEED\_STATUS  
    stm32f7xx\_hal\_conf.h, 128  
PHY\_SR  
    stm32f7xx\_hal\_conf.h, 128  
PHY\_WRITE\_TO  
    stm32f7xx\_hal\_conf.h, 128  
PI  
    Math Constants, 8  
PI2  
    Math Constants, 8  
pi\_aw\_calc  
- Integral Controllers, 9  
pi\_aw\_struct, 43  
    calc, 44  
    e, 44  
    enable, 44  
    Kaw, 44  
    Ki, 44  
    Kp, 44  
    pi\_consig, 44  
    pi\_fdb, 45  
    pi\_ffw, 45  
    pi\_int, 45  
    pi\_out, 45  
    pi\_out\_max, 45  
    pi\_out\_min, 45  
    pi\_out\_postsat, 45  
    pi\_out\_presat, 45  
    Ts, 46  
pi\_calc  
- Integral Controllers, 10  
pi\_consig  
    pi\_aw\_struct, 44  
    pi\_struct, 48  
pi\_extsat\_calc  
- Integral Controllers, 10  
pi\_fdb  
    pi\_aw\_struct, 45  
    pi\_struct, 48  
pi\_ffw  
    pi\_aw\_struct, 45  
    pi\_struct, 48  
pi\_init  
- Integral Controllers, 10  
pi\_int  
    pi\_aw\_struct, 45  
pi\_out  
    pi\_aw\_struct, 45  
    pi\_struct, 48  
pi\_out\_max  
    pi\_aw\_struct, 45  
    pi\_struct, 48  
pi\_out\_min  
    pi\_aw\_struct, 45  
    pi\_struct, 48  
pi\_struct, 46  
    calc, 47  
    e, 47  
    enable, 47  
    init, 47  
    K0, 47  
    K1, 47  
    Ki, 47  
    Kp, 47  
    pi\_consig, 48  
    pi\_fdb, 48  
    pi\_ffw, 48  
    pi\_out, 48  
    pi\_out\_max, 48  
    pi\_out\_min, 48  
    Ts, 48  
pin  
    LED, 42  
port  
    LED, 42  
PREFETCH\_ENABLE  
    stm32f7xx\_hal\_conf.h, 128  
prescaler  
    datalog\_struct, 32  
Pulses  
    step\_struct, 56  
PWM.c  
    Compare3F\_calc, 186  
    disable\_PWM, 186  
    enable\_PWM, 186  
    set\_PWM, 187  
PWM.h  
    Compare3F\_calc, 116  
    compares\_LEFT, 118  
    compares\_RIGHT, 118  
    disable\_PWM, 116  
    enable\_PWM, 117  
    set\_PWM, 117  
PWM1\_L\_GPIO\_Port  
    main.h, 92  
PWM1\_L\_Pin  
    main.h, 92  
PWM1\_R\_GPIO\_Port  
    main.h, 92  
PWM1\_R\_Pin  
    main.h, 92  
PWM2\_L\_GPIO\_Port  
    main.h, 92  
PWM2\_L\_Pin  
    main.h, 92  
PWM2\_R\_GPIO\_Port  
    main.h, 93

PWM2\_R\_Pin  
  main.h, 93  
PWM3\_L\_GPIO\_Port  
  main.h, 93  
PWM3\_L\_Pin  
  main.h, 93  
PWM3\_R\_GPIO\_Port  
  main.h, 93  
PWM3\_R\_Pin  
  main.h, 93  
PWM4\_L\_GPIO\_Port  
  main.h, 93  
PWM4\_L\_Pin  
  main.h, 93  
PWM4\_R\_GPIO\_Port  
  main.h, 93  
PWM4\_R\_Pin  
  main.h, 93  
PWM5\_L\_GPIO\_Port  
  main.h, 94  
PWM5\_L\_Pin  
  main.h, 94  
PWM5\_R\_GPIO\_Port  
  main.h, 94  
PWM5\_R\_Pin  
  main.h, 94  
PWM6\_L\_GPIO\_Port  
  main.h, 94  
PWM6\_L\_Pin  
  main.h, 94  
PWM6\_R\_GPIO\_Port  
  main.h, 94  
PWM6\_R\_Pin  
  main.h, 94

Q

  clarke3F\_struct, 29  
  iclarke3F\_struct, 37  
  irot\_struct, 41  
  rot\_struct, 54

q

  irot\_struct, 41  
  rot\_struct, 54

rampa\_calc  
  Signal Processing Functions, 18  
rampa\_dual\_calc  
  Signal Processing Functions, 19  
rampa\_dual\_struct, 49

  calc, 49  
  Decr, 49  
  enable, 49  
  in, 50  
  Incr, 50  
  out, 50

rampa\_struct, 50

  calc, 51  
  enable, 51  
  in, 51

Incr, 51  
out, 51

RMS\_calc  
  Signal Processing Functions, 19

RMS\_struct, 51

  Angle, 52  
  Angle\_ant, 52  
  Freq, 52  
  Measure, 52  
  Out\_RMS, 52  
  Sq\_Sum, 52  
  T\_exec, 53

rot\_calc  
  Clarke and Park Transformations, 14

rot\_struct, 53

  calc, 53  
  cosFi, 53  
  D, 54  
  d, 54  
  Q, 54  
  q, 54  
  sinFi, 54

SC\_det\_GPIO\_Port  
  main.h, 94

SC\_det\_Pin  
  main.h, 94

SC\_DET\_STATE  
  PCB\_IO.h, 106

set\_PWM  
  PWM.c, 187  
  PWM.h, 117

Signal Processing Functions, 16

  avg\_calc\_10\_samples, 17  
  datalog\_calc, 17  
  filtreLP\_calc, 18  
  filtreLP\_init, 18  
  N\_DATALOG, 17  
  rampa\_calc, 18  
  rampa\_dual\_calc, 19  
  RMS\_calc, 19

sinFi  
  irot\_struct, 41  
  rot\_struct, 54

SQ2  
  Math Constants, 8

SQ3  
  Math Constants, 8

Sq\_Sum  
  RMS\_struct, 52

state  
  inverterStruct, 39

Step  
  step\_struct, 56

step\_calc  
  Miscellaneous Functions, 20

step\_struct, 54

  calc, 55  
  Counter, 55

enable, 55  
 fs, 55  
 In, 55  
 Out, 55  
 Pulses, 56  
 Step, 56  
 t\_step, 56  
**stm32f7xx\_hal\_conf.h**  
 ART\_ACCELERATOR\_ENABLE, 121  
 assert\_param, 121  
 DP83848\_PHY\_ADDRESS, 122  
 ETH\_RX\_BUF\_SIZE, 122  
 ETH\_RXBUFN, 122  
 ETH\_TX\_BUF\_SIZE, 122  
 ETH\_TXBUFN, 122  
 EXTERNAL\_CLOCK\_VALUE, 122  
 HAL\_ADC\_MODULE\_ENABLED, 122  
 HAL\_CAN\_MODULE\_ENABLED, 122  
 HAL\_CORTEX\_MODULE\_ENABLED, 122  
 HAL\_DAC\_MODULE\_ENABLED, 123  
 HAL\_DMA\_MODULE\_ENABLED, 123  
 HAL\_EXTI\_MODULE\_ENABLED, 123  
 HAL\_FLASH\_MODULE\_ENABLED, 123  
 HAL\_GPIO\_MODULE\_ENABLED, 123  
 HAL\_I2C\_MODULE\_ENABLED, 123  
 HAL\_MODULE\_ENABLED, 123  
 HAL\_PWR\_MODULE\_ENABLED, 123  
 HAL\_RCC\_MODULE\_ENABLED, 123  
 HAL\_TIM\_MODULE\_ENABLED, 123  
 HSE\_STARTUP\_TIMEOUT, 124  
 HSE\_VALUE, 124  
 HSI\_VALUE, 124  
 LSE\_STARTUP\_TIMEOUT, 124  
 LSE\_VALUE, 124  
 LSI\_VALUE, 124  
 MAC\_ADDR0, 125  
 MAC\_ADDR1, 125  
 MAC\_ADDR2, 125  
 MAC\_ADDR3, 125  
 MAC\_ADDR4, 125  
 MAC\_ADDR5, 125  
 PHY\_AUTONEGO\_COMPLETE, 125  
 PHY\_AUTONEGOTIATION, 125  
 PHY\_BCR, 126  
 PHY\_BSR, 126  
 PHY\_CONFIG\_DELAY, 126  
 PHY\_DUPLEX\_STATUS, 126  
 PHY\_FULLDUPLEX\_100M, 126  
 PHY\_FULLDUPLEX\_10M, 126  
 PHY\_HALFDUPLEX\_100M, 126  
 PHY\_HALFDUPLEX\_10M, 127  
 PHY\_ISOLATE, 127  
 PHY\_JABBER\_DETECTION, 127  
 PHY\_LINKED\_STATUS, 127  
 PHY\_LOOPBACK, 127  
 PHY\_POWERDOWN, 127  
 PHY\_READ\_TO, 127  
 PHY\_RESET, 128  
**PHY\_RESET\_DELAY**, 128  
**PHY\_RESTART\_AUTONEGOTIATION**, 128  
**PHY\_SPEED\_STATUS**, 128  
**PHY\_SR**, 128  
**PHY\_WRITE\_TO**, 128  
 PREFETCH\_ENABLE, 128  
 TICK\_INT\_PRIORITY, 128  
 USE\_HAL\_ADC\_REGISTER\_CALLBACKS, 129  
 USE\_HAL\_CAN\_REGISTER\_CALLBACKS, 129  
 USE\_HAL\_CEC\_REGISTER\_CALLBACKS, 129  
 USE\_HAL\_CRYP\_REGISTER\_CALLBACKS, 129  
 USE\_HAL\_DAC\_REGISTER\_CALLBACKS, 129  
 USE\_HAL\_DCMI\_REGISTER\_CALLBACKS, 129  
 USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS, 129  
 USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS, 129  
 USE\_HAL\_DSI\_REGISTER\_CALLBACKS, 129  
 USE\_HAL\_ETH\_REGISTER\_CALLBACKS, 129  
 USE\_HAL\_HASH\_REGISTER\_CALLBACKS, 130  
 USE\_HAL\_HCD\_REGISTER\_CALLBACKS, 130  
 USE\_HAL\_I2C\_REGISTER\_CALLBACKS, 130  
 USE\_HAL\_I2S\_REGISTER\_CALLBACKS, 130  
 USE\_HAL\_IRDA\_REGISTER\_CALLBACKS, 130  
 USE\_HAL\_JPEG\_REGISTER\_CALLBACKS, 130  
 USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS, 130  
 USE\_HAL\_LTDC\_REGISTER\_CALLBACKS, 130  
 USE\_HAL\_MDIOS\_REGISTER\_CALLBACKS, 130  
 USE\_HAL\_MMIC\_REGISTER\_CALLBACKS, 130  
 USE\_HAL\_NAND\_REGISTER\_CALLBACKS, 131  
 USE\_HAL\_NOR\_REGISTER\_CALLBACKS, 131  
 USE\_HAL\_PCD\_REGISTER\_CALLBACKS, 131  
 USE\_HAL\_QSPI\_REGISTER\_CALLBACKS, 131  
 USE\_HAL\_RNG\_REGISTER\_CALLBACKS, 131  
 USE\_HAL\_RTC\_REGISTER\_CALLBACKS, 131  
 USE\_HAL\_SAI\_REGISTER\_CALLBACKS, 131  
 USE\_HAL\_SD\_REGISTER\_CALLBACKS, 131  
 USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS, 131  
 USE\_HAL\_SMARTCARD\_REGISTER\_CALLBACKS, 131  
 USE\_HAL\_SMBUS\_REGISTER\_CALLBACKS, 132  
 USE\_HAL\_SPDIFRX\_REGISTER\_CALLBACKS, 132  
 USE\_HAL\_SPI\_REGISTER\_CALLBACKS, 132  
 USE\_HAL\_SRAM\_REGISTER\_CALLBACKS, 132  
 USE\_HAL\_TIM\_REGISTER\_CALLBACKS, 132  
 USE\_HAL\_UART\_REGISTER\_CALLBACKS, 132  
 USE\_HAL\_USART\_REGISTER\_CALLBACKS, 132  
 USE\_HAL\_WWDG\_REGISTER\_CALLBACKS, 132  
 USE\_RTOS, 132  
 USE\_SPI\_CRC, 132  
 VDD\_VALUE, 133  
**stm32f7xx\_hal\_msp.c**

HAL\_MsplInit, 188  
stm32f7xx\_it.c  
    BusFault\_Handler, 190  
    CAN1\_RX0\_IRQHandler, 190  
    CAN1\_RX1\_IRQHandler, 190  
    DebugMon\_Handler, 191  
    DMA2\_Stream1\_IRQHandler, 191  
    DMA2\_Stream2\_IRQHandler, 191  
    HardFault\_Handler, 191  
    hcan1, 193  
    hdac, 193  
    hdma\_adc2, 193  
    hdma\_adc3, 193  
    htim1, 193  
    htim6, 193  
    MemManage\_Handler, 191  
    NMI\_Handler, 191  
    PendSV\_Handler, 191  
    SVC\_Handler, 192  
    SysTick\_Handler, 192  
    TIM1\_UP\_TIM10\_IRQHandler, 192  
    TIM6\_DAC\_IRQHandler, 192  
    UsageFault\_Handler, 193  
stm32f7xx\_it.h  
    BusFault\_Handler, 139  
    CAN1\_RX0\_IRQHandler, 139  
    CAN1\_RX1\_IRQHandler, 139  
    DebugMon\_Handler, 140  
    DMA2\_Stream1\_IRQHandler, 140  
    DMA2\_Stream2\_IRQHandler, 140  
    HardFault\_Handler, 140  
    MemManage\_Handler, 140  
    NMI\_Handler, 140  
    PendSV\_Handler, 140  
    SVC\_Handler, 141  
    SysTick\_Handler, 141  
    TIM1\_UP\_TIM10\_IRQHandler, 141  
    TIM6\_DAC\_IRQHandler, 141  
    UsageFault\_Handler, 142  
Stm32f7xx\_system, 21  
STM32F7xx\_System\_Private\_Defines, 23  
STM32F7xx\_System\_Private\_FunctionPrototypes, 24  
STM32F7xx\_System\_Private\_Functions, 25  
    SystemCoreClockUpdate, 25  
    SystemInit, 26  
STM32F7xx\_System\_Private\_Includes, 22  
    HSE\_VALUE, 22  
    HSI\_VALUE, 22  
STM32F7xx\_System\_Private\_Macros, 23  
STM32F7xx\_System\_Private\_TypesDefinitions, 23  
STM32F7xx\_System\_Private\_Variables, 24  
    AHBPrecTable, 24  
    APBPrecTable, 24  
    SystemCoreClock, 24  
SVC\_Handler  
    stm32f7xx\_it.c, 192  
    stm32f7xx\_it.h, 141  
svpwm\_calc  
    Utility Functions, 15  
    svpwm\_struct, 56  
        calc, 57  
        Ta, 57  
        Tb, 57  
        Tc, 57  
        valfa, 57  
        vbeta, 57  
    syscalls.c  
        \_\_attribute\_\_, 195  
        \_\_io\_getchar, 195  
        \_\_io\_putchar, 196  
        \_\_close, 196  
        \_\_execve, 196  
        \_\_exit, 196  
        \_\_fork, 196  
        \_\_fstat, 197  
        \_\_getpid, 197  
        \_\_isatty, 197  
        \_\_kill, 197  
        \_\_link, 197  
        \_\_lseek, 197  
        \_\_open, 198  
        \_\_stat, 198  
        \_\_times, 198  
        \_\_unlink, 198  
        \_\_wait, 198  
        environ, 198  
        initialise\_monitor\_handles, 198  
    sysmem.c  
        \_\_sbrk, 200  
SystemClock\_Config  
    main.c, 177  
SystemCoreClock  
    STM32F7xx\_System\_Private\_Variables, 24  
SystemCoreClockUpdate  
    STM32F7xx\_System\_Private\_Functions, 25  
SystemInit  
    STM32F7xx\_System\_Private\_Functions, 26  
SysTick\_Handler  
    stm32f7xx\_it.c, 192  
    stm32f7xx\_it.h, 141  
T\_exec  
    RMS\_struct, 53  
t\_step  
    step\_struct, 56  
Ta  
    svpwm\_struct, 57  
tasks\_1ms  
    TASKS\_1ms.c, 203  
    TASKS\_1ms.h, 144  
TASKS\_1ms.c  
    tasks\_1ms, 203  
TASKS\_1ms.h  
    tasks\_1ms, 144  
Tb  
    svpwm\_struct, 57  
Tc

svpwm\_struct, 57  
 theta\_e  
     Encoder, 34  
 theta\_m  
     Encoder, 34  
 TICK\_INT\_PRIORITY  
     stm32f7xx\_hal\_conf.h, 128  
 tim.c  
     HAL\_TIM\_Base\_MspDeInit, 205  
     HAL\_TIM\_Base\_MspInit, 205  
     HAL\_TIM\_IC\_MspDeInit, 205  
     HAL\_TIM\_IC\_MspInit, 205  
     HAL\_TIM\_MspPostInit, 205  
     htim1, 209  
     htim2, 209  
     htim4, 209  
     htim6, 209  
     htim8, 209  
     MX\_TIM1\_Init, 206  
     MX\_TIM2\_Init, 206  
     MX\_TIM4\_Init, 207  
     MX\_TIM6\_Init, 208  
     MX\_TIM8\_Init, 208  
 tim.h  
     HAL\_TIM\_MspPostInit, 146  
     htim1, 150  
     htim2, 150  
     htim4, 150  
     htim6, 150  
     htim8, 150  
     MX\_TIM1\_Init, 146  
     MX\_TIM2\_Init, 147  
     MX\_TIM4\_Init, 148  
     MX\_TIM6\_Init, 148  
     MX\_TIM8\_Init, 149  
 TIM1\_UP\_TIM10\_IRQHandler  
     stm32f7xx\_it.c, 192  
     stm32f7xx\_it.h, 141  
 TIM6\_DAC\_IRQHandler  
     stm32f7xx\_it.c, 192  
     stm32f7xx\_it.h, 141  
 Tinv\_L\_GPIO\_Port  
     main.h, 95  
 Tinv\_L\_Pin  
     main.h, 95  
 Tinv\_R\_GPIO\_Port  
     main.h, 95  
 Tinv\_R\_Pin  
     main.h, 95  
 Tmot\_L\_GPIO\_Port  
     main.h, 95  
 Tmot\_L\_Pin  
     main.h, 95  
 Tmot\_R\_GPIO\_Port  
     main.h, 95  
 Tmot\_R\_Pin  
     main.h, 95  
 TRIP\_L\_GPIO\_Port  
     main.h, 95  
     TRIP\_L\_Pin  
         main.h, 95  
     TRIP\_R\_GPIO\_Port  
         main.h, 96  
     TRIP\_R\_Pin  
         main.h, 96  
 TS  
     main.h, 96  
 Ts  
     angle\_struct, 28  
     filtreLP\_struct, 36  
     pi\_aw\_struct, 46  
     pi\_struct, 48  
 UsageFault\_Handler  
     stm32f7xx\_it.c, 193  
     stm32f7xx\_it.h, 142  
 usb\_otg.c  
     MX\_USB\_OTG\_FS\_USB\_Init, 211  
 usb\_otg.h  
     MX\_USB\_OTG\_FS\_USB\_Init, 152  
 USE\_HAL\_ADC\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 129  
 USE\_HAL\_CAN\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 129  
 USE\_HAL\_CEC\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 129  
 USE\_HAL\_CRYP\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 129  
 USE\_HAL\_DAC\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 129  
 USE\_HAL\_DCMI\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 129  
 USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 129  
 USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 129  
 USE\_HAL\_DSI\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 129  
 USE\_HAL\_ETH\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 129  
 USE\_HAL\_HASH\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 130  
 USE\_HAL\_HCD\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 130  
 USE\_HAL\_I2C\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 130  
 USE\_HAL\_I2S\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 130  
 USE\_HAL\_IRDA\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 130  
 USE\_HAL\_JPEG\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 130  
 USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 130  
 USE\_HAL\_LTDC\_REGISTER\_CALLBACKS  
     stm32f7xx\_hal\_conf.h, 130  
 USE\_HAL\_MDIOS\_REGISTER\_CALLBACKS

stm32f7xx\_hal\_conf.h, 130

USE\_HAL\_MMC\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 130

USE\_HAL\_NAND\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 131

USE\_HAL\_NOR\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 131

USE\_HAL\_PCD\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 131

USE\_HAL\_QSPI\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 131

USE\_HAL\_RNG\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 131

USE\_HAL\_RTC\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 131

USE\_HAL\_SAI\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 131

USE\_HAL\_SD\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 131

USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 131

USE\_HAL\_SMARTCARD\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 131

USE\_HAL\_SMBUS\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 132

USE\_HAL\_SPDIFRX\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 132

USE\_HAL\_SPI\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 132

USE\_HAL\_SRAM\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 132

USE\_HAL\_TIM\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 132

USE\_HAL\_UART\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 132

USE\_HAL\_USART\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 132

USE\_HAL\_WWDG\_REGISTER\_CALLBACKS  
    stm32f7xx\_hal\_conf.h, 132

USERTOS  
    stm32f7xx\_hal\_conf.h, 132

USESPI\_CRC  
    stm32f7xx\_hal\_conf.h, 132

Utility Functions, 14  
    angle\_calc, 15  
    svpwm\_calc, 15

valfa  
    svpwm\_struct, 57

valfa\_L  
    CONTROL.c, 162  
    CONTROL.h, 69

var  
    datalog\_struct, 32

vbeta  
    svpwm\_struct, 57

vbeta\_L  
    CONTROL.c, 162  
    CONTROL.h, 69

VDC  
    Measurements, 43

VDC\_L\_GPIO\_Port  
    main.h, 96

VDC\_L\_Pin  
    main.h, 96

VDC\_R\_GPIO\_Port  
    main.h, 96

VDC\_R\_Pin  
    main.h, 96

VDD\_VALUE  
    stm32f7xx\_hal\_conf.h, 133

VOLTAGE\_OFFSET  
    MEASUREMENTS.h, 102

VOLTAGE\_SLOPE  
    MEASUREMENTS.h, 102

we  
    Encoder, 34

wm\_rpm  
    Encoder, 34

WRN\_L\_GPIO\_Port  
    main.h, 96

WRN\_L\_Pin  
    main.h, 96

WRN\_R\_GPIO\_Port  
    main.h, 96

WRN\_R\_Pin  
    main.h, 97

WRN\_STATE  
    PCB\_IO.h, 107

Z  
    Encoder, 34

Z\_L\_GPIO\_Port  
    main.h, 97

Z\_L\_Pin  
    main.h, 97

Z\_R\_GPIO\_Port  
    main.h, 97

Z\_R\_Pin  
    main.h, 97