

# e-Tech Racing's Inverter Firmware

## v0

Generated by Doxygen 1.10.0



<b>1 Data Structure Index</b>	<b>1</b>
1.1 Data Structures	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Data Structure Documentation</b>	<b>5</b>
3.1 Analog Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Field Documentation	5
3.1.2.1 currentOffsets	5
3.1.2.2 ia	5
3.1.2.3 ib	6
3.1.2.4 ic	6
3.1.2.5 vDC	6
3.2 CANMessageInfo Struct Reference	6
3.2.1 Field Documentation	6
3.2.1.1 DLC	6
3.2.1.2 getSig	6
3.2.1.3 ID	7
3.2.1.4 IDE	7
3.3 Duties Struct Reference	7
3.3.1 Detailed Description	7
3.3.2 Field Documentation	7
3.3.2.1 Da	7
3.3.2.2 Db	7
3.3.2.3 Dc	8
3.4 Encoder Struct Reference	8
3.4.1 Detailed Description	8
3.4.2 Field Documentation	8
3.4.2.1 A	8
3.4.2.2 B	8
3.4.2.3 cosTheta_e	9
3.4.2.4 directionMeas	9
3.4.2.5 sinTheta_e	9
3.4.2.6 theta_e	9
3.4.2.7 we	9
3.4.2.8 Z	9
3.5 Feedback Struct Reference	9
3.5.1 Detailed Description	10
3.5.2 Field Documentation	10
3.5.2.1 idMeas	10
3.5.2.2 iqMeas	10

3.5.2.3 speedMeas	10
3.5.2.4 torqueCalc	10
3.6 InverterStruct Struct Reference	11
3.6.1 Detailed Description	11
3.6.2 Field Documentation	12
3.6.2.1 analog	12
3.6.2.2 direction	12
3.6.2.3 duties	12
3.6.2.4 enable	12
3.6.2.5 enable_pin	12
3.6.2.6 enable_port	12
3.6.2.7 enableSW	12
3.6.2.8 encoder	13
3.6.2.9 errors	13
3.6.2.10 feedback	13
3.6.2.11 hadc	13
3.6.2.12 htim	13
3.6.2.13 idLoop	13
3.6.2.14 iqLoop	13
3.6.2.15 led	13
3.6.2.16 motor	14
3.6.2.17 reference	14
3.6.2.18 speedLoop	14
3.6.2.19 state	14
3.6.2.20 tempInverter	14
3.6.2.21 tempMotor	14
3.6.2.22 vd	14
3.6.2.23 vq	14
3.6.2.24 vsMax	15
3.7 LED Struct Reference	15
3.7.1 Detailed Description	15
3.7.2 Field Documentation	15
3.7.2.1 mode	15
3.7.2.2 pin	15
3.7.2.3 port	15
3.8 MotorConstants Struct Reference	16
3.8.1 Detailed Description	16
3.8.2 Field Documentation	16
3.8.2.1 betaMinusIsc	16
3.8.2.2 eightTimesOneMinusXiSquared	16
3.8.2.3 fourTimesOneMinusXi	17
3.8.2.4 fourTimesOneMinusXiOnePlusXiSquared	17

3.8.2.5 fourTimesOneMinusXiXiSquared	17
3.8.2.6 invThreePpLambda	17
3.8.2.7 invTorqueBase	17
3.8.2.8 isc	17
3.8.2.9 lambdaDivLqMinusLd	17
3.8.2.10 oneMinusXi	17
3.8.2.11 threePpLambda	18
3.8.2.12 threePpLdMinusLq	18
3.8.2.13 torqueBase	18
3.8.2.14 twoMinusXi	18
3.8.2.15 twoMinusXiSquared	18
3.8.2.16 twoTimesOneMinusXiOnePlusXiSquared	18
3.8.2.17 twoTimesOneMinusXiXiSquared	18
3.8.2.18 xi	18
3.8.2.19 xiSquared	19
3.9 MotorParameters Struct Reference	19
3.9.1 Detailed Description	20
3.9.2 Field Documentation	20
3.9.2.1 b	20
3.9.2.2 constants	20
3.9.2.3 dTorqueMax	20
3.9.2.4 iMax	20
3.9.2.5 J	20
3.9.2.6 lambda	20
3.9.2.7 Ld	21
3.9.2.8 Lq	21
3.9.2.9 pp	21
3.9.2.10 Rs	21
3.9.2.11 speedMax_RPM	21
3.9.2.12 torqueMax	21
3.9.2.13 vDCMax	21
3.10 Reference Struct Reference	22
3.10.1 Detailed Description	22
3.10.2 Field Documentation	22
3.10.2.1 idRef	22
3.10.2.2 iqRef	22
3.10.2.3 isMaxRef	22
3.10.2.4 torqueRef	22
<b>4 File Documentation</b>	<b>23</b>
4.1 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/CAN_e-Tech.h File Reference	23
4.1.1 Detailed Description	24

4.1.2 Function Documentation	25
4.1.2.1 handle_CAN()	25
4.1.2.2 send_CAN_message()	26
4.1.3 Variable Documentation	27
4.1.3.1 enableCAN	27
4.2 CAN_e-Tech.h	27
4.3 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/CONTROL.h File Reference	27
4.3.1 Detailed Description	29
4.3.2 Function Documentation	29
4.3.2.1 calc_current_loop()	29
4.3.2.2 calc_current_reference()	29
4.3.2.3 calc_duties()	30
4.3.2.4 saturate_voltage()	31
4.4 CONTROL.h	31
4.5 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ERRORS.h File Reference	32
4.5.1 Macro Definition Documentation	33
4.5.1.1 OVERCURRENT_TH	33
4.5.1.2 OVERSPEED_TH	33
4.5.1.3 OVERTEMPERATURE_INVERTER_TH	33
4.5.1.4 OVERTEMPERATURE_MOTOR_TH	33
4.5.1.5 OVERVOLTAGE_TH	33
4.5.1.6 UNDERVOLTAGE_TH	33
4.5.2 Enumeration Type Documentation	33
4.5.2.1 InverterError	33
4.5.3 Function Documentation	34
4.5.3.1 clear_error()	34
4.5.3.2 is_error_set()	34
4.5.3.3 set_error()	35
4.6 ERRORS.h	35
4.7 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/FSM.h File Reference	36
4.7.1 Detailed Description	37
4.7.2 Function Documentation	37
4.7.2.1 eval_inv_FSM()	37
4.8 FSM.h	38
4.9 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/INVERTER.h File Reference	38
4.9.1 Detailed Description	40
4.9.2 Macro Definition Documentation	40
4.9.2.1 DT	40
4.9.2.2 TS	40
4.9.3 Enumeration Type Documentation	40
4.9.3.1 InverterState	40
4.9.4 Function Documentation	41

4.9.4.1	<a href="#">disable_control_loops()</a>	41
4.9.4.2	<a href="#">enable_control_loops()</a>	41
4.9.4.3	<a href="#">init_control_loops()</a>	41
4.9.4.4	<a href="#">initialize_inverter()</a>	42
4.9.5	<a href="#">Variable Documentation</a>	43
4.9.5.1	<a href="#">inverter_left</a>	43
4.9.5.2	<a href="#">inverter_right</a>	43
4.10	<a href="#">INVERTER.h</a>	44
4.11	<a href="#">C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/main.h File Reference</a>	44
4.11.1	<a href="#">Detailed Description</a>	47
4.11.2	<a href="#">Macro Definition Documentation</a>	47
4.11.2.1	<a href="#">A_L_GPIO_Port</a>	47
4.11.2.2	<a href="#">A_L_Pin</a>	47
4.11.2.3	<a href="#">A_R_GPIO_Port</a>	47
4.11.2.4	<a href="#">A_R_Pin</a>	48
4.11.2.5	<a href="#">B_L_GPIO_Port</a>	48
4.11.2.6	<a href="#">B_L_Pin</a>	48
4.11.2.7	<a href="#">B_R_GPIO_Port</a>	48
4.11.2.8	<a href="#">B_R_Pin</a>	48
4.11.2.9	<a href="#">DAC_GPIO_Port</a>	48
4.11.2.10	<a href="#">DAC_Pin</a>	48
4.11.2.11	<a href="#">DIR_GPIO_Port</a>	48
4.11.2.12	<a href="#">DIR_Pin</a>	48
4.11.2.13	<a href="#">ENABLE_L_GPIO_Port</a>	48
4.11.2.14	<a href="#">ENABLE_L_Pin</a>	49
4.11.2.15	<a href="#">ENABLE_R_GPIO_Port</a>	49
4.11.2.16	<a href="#">ENABLE_R_Pin</a>	49
4.11.2.17	<a href="#">ia_L_GPIO_Port</a>	49
4.11.2.18	<a href="#">ia_L_Pin</a>	49
4.11.2.19	<a href="#">ia_R_GPIO_Port</a>	49
4.11.2.20	<a href="#">ia_R_Pin</a>	49
4.11.2.21	<a href="#">ib_L_GPIO_Port</a>	49
4.11.2.22	<a href="#">ib_L_Pin</a>	49
4.11.2.23	<a href="#">ib_R_GPIO_Port</a>	49
4.11.2.24	<a href="#">ib_R_Pin</a>	50
4.11.2.25	<a href="#">ic_L_GPIO_Port</a>	50
4.11.2.26	<a href="#">ic_L_Pin</a>	50
4.11.2.27	<a href="#">ic_R_GPIO_Port</a>	50
4.11.2.28	<a href="#">ic_R_Pin</a>	50
4.11.2.29	<a href="#">LED_ERR_GPIO_Port</a>	50
4.11.2.30	<a href="#">LED_ERR_Pin</a>	50
4.11.2.31	<a href="#">LED_LEFT_GPIO_Port</a>	50

4.11.2.32 LED_LEFT_Pin . . . . .	50
4.11.2.33 LED_RIGHT_GPIO_Port . . . . .	50
4.11.2.34 LED_RIGHT_Pin . . . . .	51
4.11.2.35 PWM1_L_GPIO_Port . . . . .	51
4.11.2.36 PWM1_L_Pin . . . . .	51
4.11.2.37 PWM1_R_GPIO_Port . . . . .	51
4.11.2.38 PWM1_R_Pin . . . . .	51
4.11.2.39 PWM2_L_GPIO_Port . . . . .	51
4.11.2.40 PWM2_L_Pin . . . . .	51
4.11.2.41 PWM2_R_GPIO_Port . . . . .	51
4.11.2.42 PWM2_R_Pin . . . . .	51
4.11.2.43 PWM3_L_GPIO_Port . . . . .	51
4.11.2.44 PWM3_L_Pin . . . . .	52
4.11.2.45 PWM3_R_GPIO_Port . . . . .	52
4.11.2.46 PWM3_R_Pin . . . . .	52
4.11.2.47 PWM4_L_GPIO_Port . . . . .	52
4.11.2.48 PWM4_L_Pin . . . . .	52
4.11.2.49 PWM4_R_GPIO_Port . . . . .	52
4.11.2.50 PWM4_R_Pin . . . . .	52
4.11.2.51 PWM5_L_GPIO_Port . . . . .	52
4.11.2.52 PWM5_L_Pin . . . . .	52
4.11.2.53 PWM5_R_GPIO_Port . . . . .	52
4.11.2.54 PWM5_R_Pin . . . . .	53
4.11.2.55 PWM6_L_GPIO_Port . . . . .	53
4.11.2.56 PWM6_L_Pin . . . . .	53
4.11.2.57 PWM6_R_GPIO_Port . . . . .	53
4.11.2.58 PWM6_R_Pin . . . . .	53
4.11.2.59 SC_det_GPIO_Port . . . . .	53
4.11.2.60 SC_det_Pin . . . . .	53
4.11.2.61 Tinv_L_GPIO_Port . . . . .	53
4.11.2.62 Tinv_L_Pin . . . . .	53
4.11.2.63 Tinv_R_GPIO_Port . . . . .	53
4.11.2.64 Tinv_R_Pin . . . . .	54
4.11.2.65 Tmot_L_GPIO_Port . . . . .	54
4.11.2.66 Tmot_L_Pin . . . . .	54
4.11.2.67 Tmot_R_GPIO_Port . . . . .	54
4.11.2.68 Tmot_R_Pin . . . . .	54
4.11.2.69 TRIP_L_GPIO_Port . . . . .	54
4.11.2.70 TRIP_L_Pin . . . . .	54
4.11.2.71 TRIP_R_GPIO_Port . . . . .	54
4.11.2.72 TRIP_R_Pin . . . . .	54
4.11.2.73 VDC_L_GPIO_Port . . . . .	54



4.11.2.74 VDC_L_Pin . . . . .	55
4.11.2.75 VDC_R_GPIO_Port . . . . .	55
4.11.2.76 VDC_R_Pin . . . . .	55
4.11.2.77 WRN_L_GPIO_Port . . . . .	55
4.11.2.78 WRN_L_Pin . . . . .	55
4.11.2.79 WRN_R_GPIO_Port . . . . .	55
4.11.2.80 WRN_R_Pin . . . . .	55
4.11.2.81 Z_L_GPIO_Port . . . . .	55
4.11.2.82 Z_L_Pin . . . . .	55
4.11.2.83 Z_R_GPIO_Port . . . . .	55
4.11.2.84 Z_R_Pin . . . . .	55
4.11.3 Function Documentation . . . . .	55
4.11.3.1 Error_Handler() . . . . .	55
4.12 main.h . . . . .	56
4.13 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/MEASUREMENTS.h File Reference . . . . .	58
4.13.1 Detailed Description . . . . .	59
4.13.2 Macro Definition Documentation . . . . .	60
4.13.2.1 CURRENT_OFFSET . . . . .	60
4.13.2.2 CURRENT_SLOPE . . . . .	60
4.13.2.3 VOLTAGE_OFFSET . . . . .	60
4.13.2.4 VOLTAGE_SLOPE . . . . .	60
4.13.3 Function Documentation . . . . .	60
4.13.3.1 calibrate_offsets() . . . . .	60
4.13.3.2 get_currents_voltage() . . . . .	61
4.13.3.3 get_idiq() . . . . .	62
4.13.3.4 get_linear() . . . . .	63
4.13.3.5 get_temperature() . . . . .	64
4.13.4 Variable Documentation . . . . .	64
4.13.4.1 rawADC_left . . . . .	64
4.13.4.2 rawADC_right . . . . .	65
4.13.4.3 rawADC_temp . . . . .	65
4.13.4.4 tempInverterLUT . . . . .	65
4.13.4.5 tempMotorLUT . . . . .	65
4.14 MEASUREMENTS.h . . . . .	65
4.15 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/MOTOR.h File Reference . . . . .	66
4.15.1 Detailed Description . . . . .	67
4.15.2 Function Documentation . . . . .	68
4.15.2.1 check_motor_parameters() . . . . .	68
4.15.2.2 precalculate_motor_constants() . . . . .	69
4.15.3 Variable Documentation . . . . .	69
4.15.3.1 motor_left . . . . .	69
4.15.3.2 motor_right . . . . .	70

4.16 MOTOR.h	70
4.17 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PCB_IO.h File Reference	70
4.17.1 Detailed Description	72
4.17.2 Macro Definition Documentation	72
4.17.2.1 DIR_STATE	72
4.17.2.2 DISABLE	72
4.17.2.3 ENABLE	72
4.17.2.4 SC_DET_STATE	73
4.17.2.5 WRN_STATE	73
4.17.3 Enumeration Type Documentation	73
4.17.3.1 LEDMode	73
4.17.4 Function Documentation	73
4.17.4.1 enable_inverters()	73
4.17.4.2 handle_direction()	74
4.17.4.3 handle_LED()	74
4.17.5 Variable Documentation	75
4.17.5.1 led_left	75
4.17.5.2 led_right	75
4.17.5.3 ledError	75
4.18 PCB_IO.h	75
4.19 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PWM.h File Reference	76
4.19.1 Detailed Description	77
4.19.2 Function Documentation	77
4.19.2.1 disable_PWM()	77
4.19.2.2 enable_PWM()	78
4.19.2.3 update_PWM()	78
4.20 PWM.h	78
4.21 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/REFERENCE.h File Reference	79
4.21.1 Detailed Description	81
4.21.2 Macro Definition Documentation	81
4.21.2.1 TEMP_INVERTER_DERATING	81
4.21.2.2 TEMP_INVERTER_MAX	81
4.21.2.3 TEMP_MOTOR_DERATING	81
4.21.2.4 TEMP_MOTOR_MAX	81
4.21.3 Function Documentation	81
4.21.3.1 calculate_derated_current()	81
4.21.3.2 derate_current_reference()	82
4.21.3.3 handle_torqueRef()	83
4.21.3.4 limit_torque_to_prevent_overspeed()	84
4.21.3.5 saturate_symmetric()	85
4.21.3.6 set_torque_direction()	85
4.21.4 Variable Documentation	86

4.21.4.1 torqueRefIn_left . . . . .	86
4.21.4.2 torqueRefIn_right . . . . .	86
4.22 REFERENCE.h . . . . .	86
4.23 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/TASKS_1ms.h File Reference . . . . .	87
4.23.1 Detailed Description . . . . .	88
4.23.2 Function Documentation . . . . .	88
4.23.2.1 handle_oversampling_faults() . . . . .	88
4.23.2.2 read_temperatures() . . . . .	89
4.23.2.3 tasks_1ms() . . . . .	90
4.24 TASKS_1ms.h . . . . .	91
4.25 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/TASKS_CRITICAL.h File Reference . . . . .	91
4.25.1 Detailed Description . . . . .	92
4.25.2 Function Documentation . . . . .	92
4.25.2.1 tasks_critical_left() . . . . .	92
4.25.2.2 tasks_critical_right() . . . . .	93
4.26 TASKS_CRITICAL.h . . . . .	93
4.27 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/CAN_e-Tech.c File Reference . . . . .	93
4.27.1 Detailed Description . . . . .	94
4.27.2 Function Documentation . . . . .	95
4.27.2.1 handle_CAN() . . . . .	95
4.27.2.2 send_CAN_message() . . . . .	96
4.27.3 Variable Documentation . . . . .	97
4.27.3.1 keepAlive . . . . .	97
4.28 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/CONTROL.c File Reference . . . . .	97
4.28.1 Detailed Description . . . . .	98
4.28.2 Function Documentation . . . . .	98
4.28.2.1 calc_current_loop() . . . . .	98
4.28.2.2 calc_current_reference() . . . . .	99
4.28.2.3 calc_duties() . . . . .	99
4.28.2.4 saturate_voltage() . . . . .	100
4.29 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ERRORS.c File Reference . . . . .	100
4.29.1 Detailed Description . . . . .	101
4.29.2 Function Documentation . . . . .	102
4.29.2.1 clear_error() . . . . .	102
4.29.2.2 is_error_set() . . . . .	103
4.29.2.3 set_error() . . . . .	103
4.30 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/FSM.c File Reference . . . . .	104
4.30.1 Detailed Description . . . . .	105
4.30.2 Function Documentation . . . . .	105
4.30.2.1 eval_inv_FSM() . . . . .	105
4.31 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/INVERTER.c File Reference . . . . .	105
4.31.1 Detailed Description . . . . .	106

4.31.2 Function Documentation	107
4.31.2.1 disable_control_loops()	107
4.31.2.2 enable_control_loops()	107
4.31.2.3 init_control_loops()	107
4.31.2.4 initialize_inverter()	108
4.31.3 Variable Documentation	109
4.31.3.1 inverter_left	109
4.31.3.2 inverter_right	109
4.32 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/main.c File Reference	109
4.32.1 Detailed Description	110
4.32.2 Function Documentation	110
4.32.2.1 Error_Handler()	110
4.32.2.2 main()	111
4.32.2.3 SystemClock_Config()	111
4.33 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/MEASUREMENTS.c File Reference	112
4.33.1 Detailed Description	122
4.33.2 Function Documentation	123
4.33.2.1 calibrate_offsets()	123
4.33.2.2 get_currents_voltage()	123
4.33.2.3 get_idiq()	124
4.33.2.4 get_linear()	125
4.33.2.5 get_temperature()	126
4.33.3 Variable Documentation	126
4.33.3.1 rawADC_left	126
4.33.3.2 rawADC_right	126
4.33.3.3 rawADC_temp	127
4.33.3.4 tempInverterLUT	127
4.33.3.5 tempMotorLUT	132
4.34 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/MOTOR.c File Reference	137
4.34.1 Detailed Description	138
4.34.2 Function Documentation	139
4.34.2.1 check_motor_parameters()	139
4.34.2.2 precalculate_motor_constants()	139
4.34.3 Variable Documentation	140
4.34.3.1 motor_left	140
4.34.3.2 motor_right	140
4.35 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/PCB_IO.c File Reference	141
4.35.1 Detailed Description	142
4.35.2 Function Documentation	142
4.35.2.1 enable_inverters()	142
4.35.2.2 handle_direction()	142
4.35.2.3 handle_LED()	143

4.35.3 Variable Documentation . . . . .	144
4.35.3.1 led_left . . . . .	144
4.35.3.2 led_right . . . . .	144
4.35.3.3 ledError . . . . .	144
4.36 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/PWM.c File Reference . . . . .	144
4.36.1 Detailed Description . . . . .	145
4.36.2 Function Documentation . . . . .	145
4.36.2.1 disable_PWM() . . . . .	145
4.36.2.2 enable_PWM() . . . . .	145
4.36.2.3 update_PWM() . . . . .	145
4.37 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/REFERENCE.c File Reference . . . . .	146
4.37.1 Detailed Description . . . . .	148
4.37.2 Function Documentation . . . . .	148
4.37.2.1 calculate_derated_current() . . . . .	148
4.37.2.2 derate_current_reference() . . . . .	148
4.37.2.3 handle_torqueRef() . . . . .	149
4.37.2.4 limit_torque_to_prevent_overspeed() . . . . .	150
4.37.2.5 saturate_symmetric() . . . . .	151
4.37.2.6 set_torque_direction() . . . . .	152
4.37.3 Variable Documentation . . . . .	152
4.37.3.1 torqueRefIn_left . . . . .	152
4.37.3.2 torqueRefIn_right . . . . .	152
4.38 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/stm32f7xx_it.c File Reference . . . . .	152
4.38.1 Detailed Description . . . . .	154
4.38.2 Function Documentation . . . . .	154
4.38.2.1 BusFault_Handler() . . . . .	154
4.38.2.2 CAN1_RX0_IRQHandler() . . . . .	154
4.38.2.3 DebugMon_Handler() . . . . .	155
4.38.2.4 DMA2_Stream0_IRQHandler() . . . . .	155
4.38.2.5 DMA2_Stream1_IRQHandler() . . . . .	155
4.38.2.6 DMA2_Stream2_IRQHandler() . . . . .	155
4.38.2.7 HardFault_Handler() . . . . .	156
4.38.2.8 MemManage_Handler() . . . . .	156
4.38.2.9 NMI_Handler() . . . . .	156
4.38.2.10 PendSV_Handler() . . . . .	156
4.38.2.11 SVC_Handler() . . . . .	156
4.38.2.12 SysTick_Handler() . . . . .	156
4.38.2.13 TIM1_UP_TIM10_IRQHandler() . . . . .	156
4.38.2.14 TIM6_DAC_IRQHandler() . . . . .	157
4.38.2.15 UsageFault_Handler() . . . . .	157
4.38.3 Variable Documentation . . . . .	157
4.38.3.1 hcan1 . . . . .	157

---

4.38.3.2 hdac . . . . .	157
4.38.3.3 hdma_adc1 . . . . .	157
4.38.3.4 hdma_adc2 . . . . .	158
4.38.3.5 hdma_adc3 . . . . .	158
4.38.3.6 htim1 . . . . .	158
4.38.3.7 htim6 . . . . .	158
4.39 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/TASKS_1ms.c File Reference . . . . .	158
4.39.1 Detailed Description . . . . .	159
4.39.2 Function Documentation . . . . .	159
4.39.2.1 handle_overtemperature_faults() . . . . .	159
4.39.2.2 read_temperatures() . . . . .	160
4.39.2.3 tasks_1ms() . . . . .	160
4.40 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/TASKS_CRITICAL.c File Reference . . . . .	161
4.40.1 Detailed Description . . . . .	162
4.40.2 Function Documentation . . . . .	163
4.40.2.1 tasks_critical_left() . . . . .	163
4.40.2.2 tasks_critical_right() . . . . .	163
4.40.3 Variable Documentation . . . . .	164
4.40.3.1 angle_left . . . . .	164
4.40.3.2 elapsed_ticks . . . . .	164
4.40.3.3 freqRamp_left . . . . .	164
4.40.3.4 start_ticks . . . . .	164
<b>Index</b>	<b>165</b>

# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">Analog</a>		
	Structure for ADC measurements in units . . . . .	5
<a href="#">CANMessageInfo</a>	. . . . .	6
<a href="#">Duties</a>		
	Structure to hold PWM configuration parameters . . . . .	7
<a href="#">Encoder</a>		
	Structure for encoder reading . . . . .	8
<a href="#">Feedback</a>		
	Structure for feedback values . . . . .	9
<a href="#">InverterStruct</a>		
	Inverter structure . . . . .	11
<a href="#">LED</a>		
	LED structure . . . . .	15
<a href="#">MotorConstants</a>		
	Structure to hold precomputed motor constants . . . . .	16
<a href="#">MotorParameters</a>		
	Structure to hold motor parameters . . . . .	19
<a href="#">Reference</a>		
	Structure for reference values . . . . .	22





## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/CAN_e-Tech.h	
Header file for handling CAN communication with the car . . . . .	23
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/CONTROL.h	
Header file for the control loop . . . . .	27
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ERRORS.h	
Header for the inverter Finite State Machine . . . . .	32
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/FSM.h	
Header for the inverter Finite State Machine . . . . .	36
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/INVERTER.h	
Header file for the inverter struct and extern variables . . . . .	38
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/main.h	
: Header for main.c file. This file contains the common defines of the application . . . . .	44
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/MEASUREMENTS.h	
Header file for handling measurements . . . . .	58
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/MOTOR.h	
Header file for motor parameters . . . . .	66
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PCB_IO.h	
Header file for handling GPIOs . . . . .	70
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PWM.h	
Header file for controlling PWM output . . . . .	76
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/REFERENCE.h	
Header file for torque reference handling . . . . .	79
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/TASKS_1ms.h	
Header file for functions related to tasks executed every 1ms . . . . .	87
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/TASKS_CRITICAL.h	
Header file for functions related to tasks executed in each PWM timer interruption . . . . .	91
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/CAN_e-Tech.c	
This file contains functions to handle CAN communication with the car . . . . .	93
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/CONTROL.c	
This file provides code for the control loop . . . . .	97
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ERRORS.c	
Header file for the necessary components to set, read and clear ERRORS . . . . .	100
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/FSM.c	
This file provides code for the inverter Finite State Machine . . . . .	104
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/INVERTER.c	
This file provides code for the inverter struct . . . . .	105

C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">main.c</a>	
: Main program body . . . . .	109
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">MEASUREMENTS.c</a>	
This file provides functions for handling measurements . . . . .	112
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">MOTOR.c</a>	
Source file for motor parameters . . . . .	137
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">PCB_IO.c</a>	
This file provides functions for handling GPIOs . . . . .	141
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">PWM.c</a>	
This file provides functions for controlling PWM output . . . . .	144
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">REFERENCE.c</a>	
Source file for torque reference handling . . . . .	146
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">stm32f7xx_it.c</a>	
Interrupt Service Routines . . . . .	152
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">TASKS_1ms.c</a>	
This file contains functions to execute tasks every 1ms . . . . .	158
C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ <a href="#">TASKS_CRITICAL.c</a>	
This file contains functions executed in each PWM timer interruption . . . . .	161

## Chapter 3

# Data Structure Documentation

### 3.1 Analog Struct Reference

Structure for ADC measurements in units.

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

#### Data Fields

- float [ia](#)
- float [ib](#)
- float [ic](#)
- float [vDC](#)
- float [currentOffsets](#) [3]

#### 3.1.1 Detailed Description

Structure for ADC measurements in units.

#### 3.1.2 Field Documentation

##### 3.1.2.1 currentOffsets

```
float currentOffsets[3]
```

Offsets for the current measurements

##### 3.1.2.2 ia

```
float ia
```

Phase A current in A

### 3.1.2.3 **ib**

```
float ib
```

Phase B current in A

### 3.1.2.4 **ic**

```
float ic
```

Phase C current in A

### 3.1.2.5 **vDC**

```
float vDC
```

DC link voltage in V

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

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

## 3.2 CANMessageInfo Struct Reference

```
#include <CAN_e-Tech.h>
```

### Data Fields

- const uint32\_t [ID](#)
- const uint8\_t [IDE](#)
- const uint8\_t [DLC](#)
- const signal\_positioned \* [getSig](#)

### 3.2.1 Field Documentation

#### 3.2.1.1 **DLC**

```
const uint8_t DLC
```

#### 3.2.1.2 **getSig**

```
const signal_positioned* getSig
```

### 3.2.1.3 ID

```
const uint32_t ID
```

### 3.2.1.4 IDE

```
const uint8_t IDE
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[CAN\\_e-Tech.h](#)

## 3.3 Duties Struct Reference

Structure to hold PWM configuration parameters.

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

### Data Fields

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

### 3.3.1 Detailed Description

Structure to hold PWM configuration parameters.

### 3.3.2 Field Documentation

#### 3.3.2.1 Da

```
float Da
```

Duty cycle for channel 1

#### 3.3.2.2 Db

```
float Db
```

Duty cycle for channel 2

### 3.3.2.3 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](#)

## 3.4 Encoder Struct Reference

Structure for encoder reading.

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

### Data Fields

- `uint16_t` [A](#)
- `uint16_t` [B](#)
- `uint16_t` [Z](#)
- `float` [we](#)
- `float` [theta\\_e](#)
- `float` [sinTheta\\_e](#)
- `float` [cosTheta\\_e](#)
- `uint8_t` [directionMeas](#)

### 3.4.1 Detailed Description

Structure for encoder reading.

### 3.4.2 Field Documentation

#### 3.4.2.1 A

`uint16_t` [A](#)

[Encoder](#) channel A value

#### 3.4.2.2 B

`uint16_t` [B](#)

[Encoder](#) channel B value

#### 3.4.2.3 cosTheta\_e

```
float cosTheta_e
```

Electrical rotor position cosine

#### 3.4.2.4 directionMeas

```
uint8_t directionMeas
```

Measured direction

#### 3.4.2.5 sinTheta\_e

```
float sinTheta_e
```

Electrical rotor position sine

#### 3.4.2.6 theta\_e

```
float theta_e
```

Electrical rotor position

#### 3.4.2.7 we

```
float we
```

Electrical angular velocity

#### 3.4.2.8 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](#)

## 3.5 Feedback Struct Reference

Structure for feedback values.

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

## Data Fields

- float [idMeas](#)
- float [iqMeas](#)
- float [torqueCalc](#)
- float [speedMeas](#)

### 3.5.1 Detailed Description

Structure for feedback values.

### 3.5.2 Field Documentation

#### 3.5.2.1 idMeas

```
float idMeas
```

Measured d-axis current in A

#### 3.5.2.2 iqMeas

```
float iqMeas
```

Measured q-axis current in A

#### 3.5.2.3 speedMeas

```
float speedMeas
```

Measured speed in RPM

#### 3.5.2.4 torqueCalc

```
float torqueCalc
```

Calculated torque in N·m

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

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

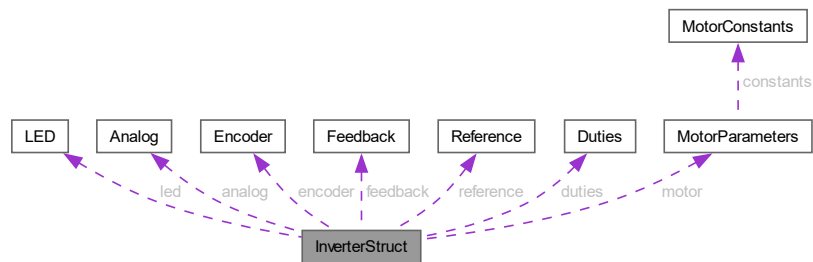


## 3.6 InverterStruct Struct Reference

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](#)
- [Analog](#) [analog](#)
- [Encoder](#) [encoder](#)
- [Feedback](#) [feedback](#)
- [Reference](#) [reference](#)
- [Duties](#) [duties](#)
- int8\_t [direction](#)
- float [templInverter](#)
- float [tempMotor](#)
- [MotorParameters](#) \* [motor](#)
- pi\_struct [idLoop](#)
- pi\_struct [iqLoop](#)
- float [vsMax](#)
- float [vd](#)
- float [vq](#)
- pi\_struct [speedLoop](#)
- [InverterError](#) [errors](#)
- bool [enable](#)
- bool [enableSW](#)

### 3.6.1 Detailed Description

Inverter structure.

### 3.6.2 Field Documentation

#### 3.6.2.1 analog

`Analog` analog

Structure for phase currents and DC voltage measurements

#### 3.6.2.2 direction

`int8_t` direction

Motor direction: 1 CW, -1 CCW, 0 stopped

#### 3.6.2.3 duties

`Duties` duties

Structure for duty cycles for phases A, B, and C

#### 3.6.2.4 enable

`bool` enable

Enable bit for transitioning states

#### 3.6.2.5 enable\_pin

`uint16_t` enable\_pin

Pin number for enabling/disabling the inverter

#### 3.6.2.6 enable\_port

`GPIO_TypeDef*` enable\_port

Pointer to GPIO port for enabling/disabling the inverter

#### 3.6.2.7 enableSW

`bool` enableSW

External enable order (needs HW enable to set inv.enable to 1, and if the FAULT state is entered, enableSW must be set to 0 to transition to the IDLE state)

### 3.6.2.8 encoder

`Encoder` encoder

Structure for encoder input

### 3.6.2.9 errors

`InverterError` errors

Error field storing error bits, using InverterError enum

### 3.6.2.10 feedback

`Feedback` feedback

Structure for measured currents and calculated mechanical torque and speed

### 3.6.2.11 hadc

`ADC_HandleTypeDef*` hadc

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

### 3.6.2.12 htim

`TIM_HandleTypeDef*` htim

Handle of the timer peripheral for PWM output

### 3.6.2.13 idLoop

`pi_struct` idLoop

PI controller for d-axis current

### 3.6.2.14 iqLoop

`pi_struct` iqLoop

PI controller for q-axis current

### 3.6.2.15 led

`LED*` led

Pointer to `LED` control structure

### 3.6.2.16 motor

`MotorParameters* motor`

Motor parameters struct

### 3.6.2.17 reference

`Reference reference`

Structure for referece currents and torque

### 3.6.2.18 speedLoop

`pi_struct speedLoop`

PI controller for motor speed

### 3.6.2.19 state

`InverterState state`

Current state of inverter operation

### 3.6.2.20 tempInverter

`float tempInverter`

Semiconductor temperature in degC

### 3.6.2.21 tempMotor

`float tempMotor`

Motor temperature in degC

### 3.6.2.22 vd

`float vd`

d-axis voltage

### 3.6.2.23 vq

`float vq`

q-axis voltage

### 3.6.2.24 vsMax

```
float vsMax
```

Maximum output voltage, should be calculated as  $v_{DC} / \sqrt{3}$  in volts

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

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

## 3.7 LED Struct Reference

[LED](#) structure.

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

### Data Fields

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

### 3.7.1 Detailed Description

[LED](#) structure.

### 3.7.2 Field Documentation

#### 3.7.2.1 mode

[LEDMode](#) [mode](#)

Current [LED](#) mode

#### 3.7.2.2 pin

uint16\_t [pin](#)

Pin number for controlling the [LED](#)

#### 3.7.2.3 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](#)

## 3.8 MotorConstants Struct Reference

Structure to hold precomputed motor constants.

```
#include <MOTOR.h>
```

### Data Fields

- float [threePpLambda](#)
- float [threePpLdMinusLq](#)
- float [invThreePpLambda](#)
- float [isc](#)
- float [torqueBase](#)
- float [invTorqueBase](#)
- float [xi](#)
- float [xiSquared](#)
- float [oneMinusXi](#)
- float [twoMinusXi](#)
- float [fourTimesOneMinusXi](#)
- float [eightTimesOneMinusXiSquared](#)
- float [twoMinusXiSquared](#)
- float [twoTimesOneMinusXiOnePlusXiSquared](#)
- float [twoTimesOneMinusXiXiSquared](#)
- float [fourTimesOneMinusXiOnePlusXiSquared](#)
- float [fourTimesOneMinusXiXiSquared](#)
- float [lambdaDivLqMinusLd](#)
- float [betaMinusIsc](#)

### 3.8.1 Detailed Description

Structure to hold precomputed motor constants.

### 3.8.2 Field Documentation

#### 3.8.2.1 [betaMinusIsc](#)

```
float betaMinusIsc
```

$$\lambda / (L_q - L_d) - \lambda / L_d$$

#### 3.8.2.2 [eightTimesOneMinusXiSquared](#)

```
float eightTimesOneMinusXiSquared
```

$$8 * (1 - L_q / L_d)^2$$

### 3.8.2.3 fourTimesOneMinusXi

```
float fourTimesOneMinusXi
```

$$4 * (1 - Lq / Ld)$$

### 3.8.2.4 fourTimesOneMinusXiOnePlusXiSquared

```
float fourTimesOneMinusXiOnePlusXiSquared
```

$$4 * (1 - Lq / Ld) * (1 + (Lq / Ld)^2)$$

### 3.8.2.5 fourTimesOneMinusXiXiSquared

```
float fourTimesOneMinusXiXiSquared
```

$$4 * (1 - Lq / Ld) * (Lq / Ld)^2$$

### 3.8.2.6 invThreePpLambda

```
float invThreePpLambda
```

$$1 / (3 * pp * lambda)$$

### 3.8.2.7 invTorqueBase

```
float invTorqueBase
```

$$1 / (3 * pp * lambda^2 / Ld)$$

### 3.8.2.8 isc

```
float isc
```

$$-lambda / Ld$$

### 3.8.2.9 lambdaDivLqMinusLd

```
float lambdaDivLqMinusLd
```

$$lambda / (Lq - Ld)$$

### 3.8.2.10 oneMinusXi

```
float oneMinusXi
```

$$1 - Lq / Ld$$

**3.8.2.11 threePpLambda**

```
float threePpLambda
```

$$3 * pp * lambda$$

**3.8.2.12 threePpLdMinusLq**

```
float threePpLdMinusLq
```

$$3 * pp * (Ld - Lq)$$

**3.8.2.13 torqueBase**

```
float torqueBase
```

$$3 * pp * lambda^2 / Ld$$

**3.8.2.14 twoMinusXi**

```
float twoMinusXi
```

$$2 - Lq / Ld$$

**3.8.2.15 twoMinusXiSquared**

```
float twoMinusXiSquared
```

$$(2 - Lq / Ld)^2$$

**3.8.2.16 twoTimesOneMinusXiOnePlusXiSquared**

```
float twoTimesOneMinusXiOnePlusXiSquared
```

$$2 * (1 - Lq / Ld) * (1 + (Lq / Ld)^2)$$

**3.8.2.17 twoTimesOneMinusXiXiSquared**

```
float twoTimesOneMinusXiXiSquared
```

$$2 * (1 - Lq / Ld) * (Lq / Ld)^2$$

**3.8.2.18 xi**

```
float xi
```

$$Lq / Ld$$



## 3.8.2.19 xiSquared

```
float xiSquared
```

$$(L_q / L_d)^2$$

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

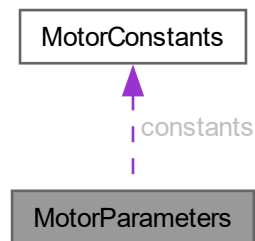
- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/[MOTOR.h](#)

## 3.9 MotorParameters Struct Reference

Structure to hold motor parameters.

```
#include <MOTOR.h>
```

Collaboration diagram for MotorParameters:



## Data Fields

- float [Ld](#)
- float [Lq](#)
- float [Rs](#)
- float [lambda](#)
- uint8\_t [pp](#)
- float [J](#)
- float [b](#)
- float [torqueMax](#)
- float [dTorqueMax](#)
- float [speedMax\\_RPM](#)
- float [iMax](#)
- float [vDCMax](#)
- [MotorConstants constants](#)

### 3.9.1 Detailed Description

Structure to hold motor parameters.

### 3.9.2 Field Documentation

#### 3.9.2.1 **b**

`float b`

Viscous friction in N·m·s

#### 3.9.2.2 **constants**

`MotorConstants` constants

Precomputed motor constants

#### 3.9.2.3 **dTorqueMax**

`float dTorqueMax`

Maximum torque increment in N·m/s

#### 3.9.2.4 **iMax**

`float iMax`

Maximum phase current (peak value, or RMS\*sqrt2)

#### 3.9.2.5 **J**

`float J`

Rotational inertia in N·m·s<sup>2</sup>

#### 3.9.2.6 **lambda**

`float lambda`

Magnet flux linkage measured  $V_{pk\_ph-n} \cdot s$  (phase-neutral peak voltage divided by electrical speed in rad/s)

### 3.9.2.7 Ld

```
float Ld
```

D-axis inductance in Henries

### 3.9.2.8 Lq

```
float Lq
```

Q-axis inductance in Henries

### 3.9.2.9 pp

```
uint8_t pp
```

Pole pairs (total number of poles divided by 2)

### 3.9.2.10 Rs

```
float Rs
```

Stator resistance in Ohms

### 3.9.2.11 speedMax\_RPM

```
float speedMax_RPM
```

Maximum speed in RPM

### 3.9.2.12 torqueMax

```
float torqueMax
```

Maximum torque in N·m

### 3.9.2.13 vDCMax

```
float vDCMax
```

Maximum DC bus voltage in volts

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

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

## 3.10 Reference Struct Reference

Structure for reference values.

```
#include <REFERENCE.h>
```

### Data Fields

- float [idRef](#)
- float [iqRef](#)
- float [isMaxRef](#)
- float [torqueRef](#)

### 3.10.1 Detailed Description

Structure for reference values.

### 3.10.2 Field Documentation

#### 3.10.2.1 idRef

```
float idRef
```

[Reference](#) d-axis current in A

#### 3.10.2.2 iqRef

```
float iqRef
```

[Reference](#) q-axis current in A

#### 3.10.2.3 isMaxRef

```
float isMaxRef
```

Maximum reference current in A

#### 3.10.2.4 torqueRef

```
float torqueRef
```

[Reference](#) torque in N·m

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

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

## Chapter 4

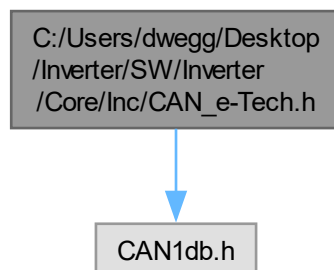
# File Documentation

### 4.1 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/CAN\_e-Tech.h File Reference

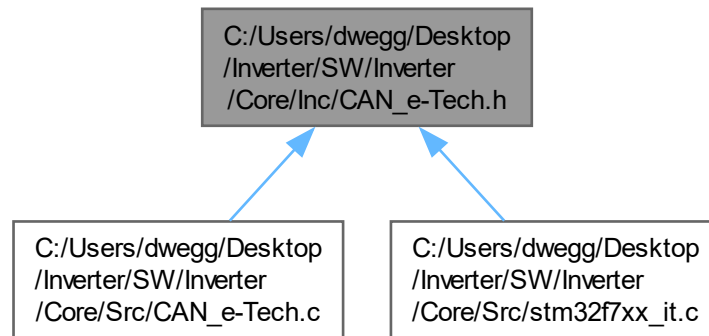
Header file for handling CAN communication with the car.

```
#include "CAN1db.h"
```

Include dependency graph for CAN\_e-Tech.h:



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



## Data Structures

- struct [CANMessageInfo](#)

## Functions

- void [handle\\_CAN](#) (CAN\_HandleTypeDef \*hcan)  
*Handle CAN messages.*
- void [send\\_CAN\\_message](#) (CAN\_HandleTypeDef \*hcan, void \*dbc\_msg, const float \*data)  
*Send a CAN message using CAN1db.h information.*

## Variables

- uint8\_t [enableCAN](#)

### 4.1.1 Detailed Description

Header file for handling CAN communication with the car.

#### 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>

## 4.1.2 Function Documentation

### 4.1.2.1 handle\_CAN()

```
void handle_CAN (
    CAN_HandleTypeDef * hcan )
```

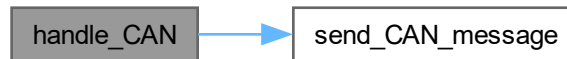
Handle CAN messages.

This function implements the logic to handle received CAN messages.

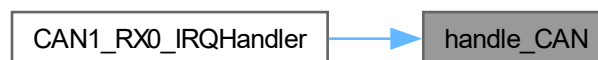
## Parameters

<i>hcan</i>	Pointer to the CAN handle structure.
-------------	--------------------------------------

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.1.2.2 send\_CAN\_message()

```
void send_CAN_message (
    CAN_HandleTypeDef * hcan,
    void * dbc_msg,
    const float * data )
```

Send a CAN message using CAN1db.h information.

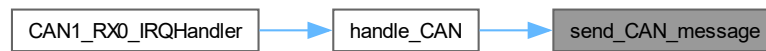
This function prepares and sends a CAN message using information from CAN1db.h.

## Parameters

<i>hcan</i>	Pointer to the CAN handle structure.
<i>dbc_msg</i>	Pointer to the structure containing CAN message information from CAN1db.h.
<i>data</i>	Pointer to the array of float data to be sent.



Here is the caller graph for this function:



### 4.1.3 Variable Documentation

#### 4.1.3.1 enableCAN

```
uint8_t enableCAN [extern]
```

## 4.2 CAN\_e-Tech.h

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

```

00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020 #ifndef CAN_E_TECH_H
00021 #define CAN_E_TECH_H
00022
00023 #include "CAN1db.h" // needs the CAN1db and its types
00024
00025 extern uint8_t enableCAN;
00026
00027 typedef struct {
00028     const uint32_t ID;
00029     const uint8_t IDE;
00030     const uint8_t DLC;
00031     const signal_positioned *getSig;
00032 } CANMessageInfo;
00033
00041 void handle_CAN(CAN_HandleTypeDef *hcan);
00042
00052 void send_CAN_message(CAN_HandleTypeDef *hcan, void *dbc_msg, const float *data);
00053
00054 #endif /* CAN_E_TECH_H */

```

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

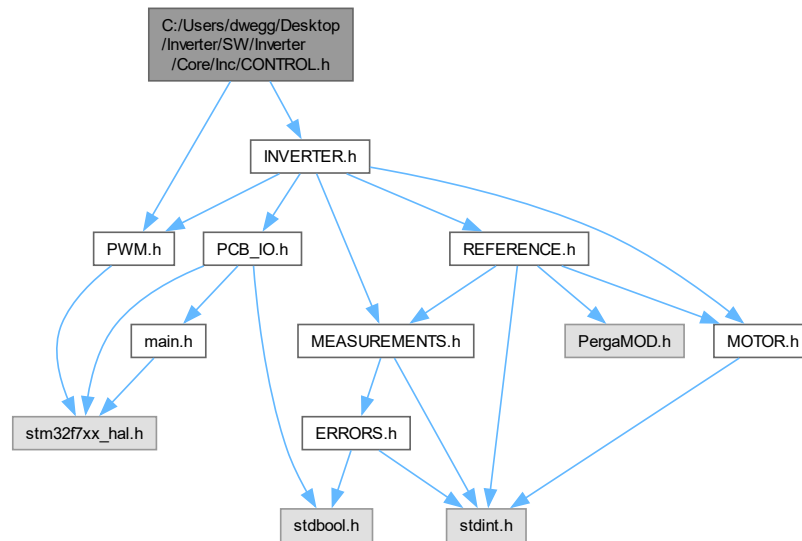
Header file for the control loop.

```

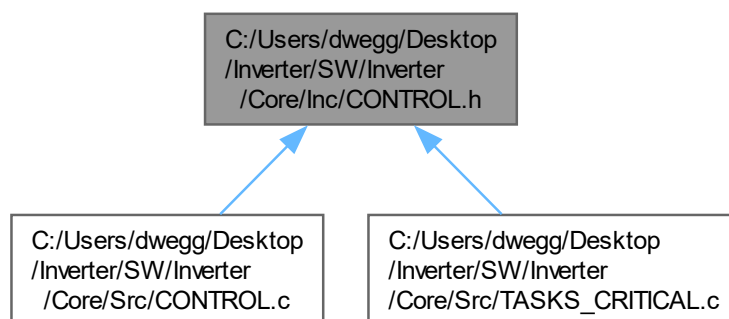
#include "PWM.h"
#include "INVERTER.h"

```

Include dependency graph for CONTROL.h:



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



## Functions

- void `calc_current_reference` (`MotorParameters` \*motor, volatile `Reference` \*reference)

*Calculates the current references using a FOC algorithm. It computes the current vector for the MTPA trajectory and limits the current reference to `isMaxRef` (calculated by derating, starting from the motor's maximum current). The MTPV trajectory is not implemented to save some computation time due to the nature of the motors expected. In order to implement field weakening, an external voltage loop modifying `gammaRef` is needed and should be called inside here. When implementing field weakening, special attention must be put to the torque reference being near 0 or differing from the speed sign (regeneration). A minimum id current must be set for speeds higher than  $V_s/\lambda$ . Study thoroughly, simulate first.*

- void `calc_current_loop` (volatile `InverterStruct` \*inv)

*Calculates the id-iq loops.*

- void `saturate_voltage` (volatile `InverterStruct` \*inv)  
*Saturates PI output to not surpass DC voltage.*
- void `calc_duties` (float vd, float vq, float vDC, float sinTheta\_e, float cosTheta\_e, volatile `Duties` \*duties)  
*function.*

### 4.3.1 Detailed Description

Header file for the control loop.

Attention

Copyright (c) 2024 David Redondo (@dweggg on 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>

### 4.3.2 Function Documentation

#### 4.3.2.1 `calc_current_loop()`

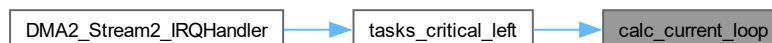
```
void calc_current_loop (
    volatile InverterStruct * inv )
```

Calculates the id-iq loops.

Parameters

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

Here is the caller graph for this function:



#### 4.3.2.2 `calc_current_reference()`

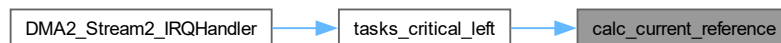
```
void calc_current_reference (
    MotorParameters * motor,
    volatile Reference * reference )
```

Calculates the current references using a FOC algorithm. It computes the current vector for the MTPA trajectory and limits the current reference to `isMaxRef` (calculated by derating, starting from the motor's maximum current). The MTPV trajectory is not implemented to save some computation time due to the nature of the motors expected. In order to implement field weakening, an external voltage loop modifying `gammaRef` is needed and should be called inside here. When implementing field weakening, special attention must be put to the torque reference being near 0 or differing from the speed sign (regeneration). A minimum id current must be set for speeds higher than  $V_s/\lambda$ . Study thoroughly, simulate first.

#### Parameters

in	<i>motor</i>	Pointer to the motor parameters structure.
in, out	<i>reference</i>	Pointer to the reference struct.

Here is the caller graph for this function:



#### 4.3.2.3 calc\_duties()

```

void calc_duties (
    float vd,
    float vq,
    float vDC,
    float sinTheta_e,
    float cosTheta_e,
    volatile Duties * duties )
  
```

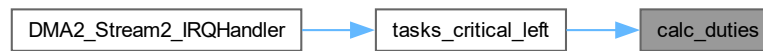
function.

This function calculates the inverse Park transform and the duty cycles using SVPWM

#### Parameters

in	<i>vd</i>	Voltage in the d-axis.
in	<i>vq</i>	Voltage in the q-axis.
in	<i>vDC</i>	DC voltage.
in	<i>sinTheta_e</i>	Electrical angle sine (-1..1)
in	<i>cosTheta_e</i>	Electrical angle cosine (-1..1)
out	<i>duties</i>	Pointer to the duties structure.

Here is the caller graph for this function:



#### 4.3.2.4 saturate\_voltage()

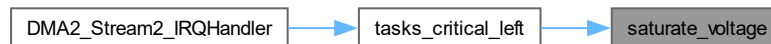
```
void saturate_voltage (
    volatile InverterStruct * inv )
```

Saturates PI output to not surpass DC voltage.

##### Parameters

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

Here is the caller graph for this function:



## 4.4 CONTROL.h

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

```

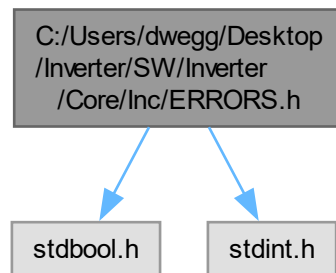
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020 #ifndef CONTROL_H
00021 #define CONTROL_H
00022
00023 #include "PWM.h" // duties struct
00024 #include "INVERTER.h" // TS & Inverter struct
00025
00041 void calc_current_reference(MotorParameters * motor, volatile Reference * reference);
00042
00048 void calc_current_loop(volatile InverterStruct *inv);
00049
00055 void saturate_voltage(volatile InverterStruct *inv);
00056
00069 void calc_duties(float vd, float vq, float vDC, float sinTheta_e, float cosTheta_e, volatile Duties
    *duties);
00070
00071 #endif /* CONTROL_H */
```

## 4.5 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/ERRORS.h File Reference

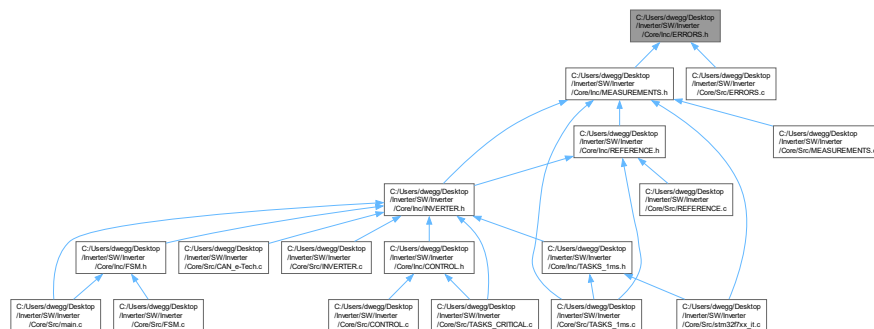
```
#include <stdbool.h>
```

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

Include dependency graph for ERRORS.h:



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



### Macros

- #define **OVERTEMPERATURE\_INVERTER\_TH** 60.0f
- #define **OVERVOLTAGE\_TH** 600.0f
- #define **OVERCURRENT\_TH** 100.0f
- #define **OVERSPEED\_TH** 20000.0f
- #define **UNDERVOLTAGE\_TH** 10.0f
- #define **OVERTEMPERATURE\_MOTOR\_TH** 90.0f

### Enumerations

- enum **InverterError** {  
**NONE** = 0 , **POWER\_FAULT** = (1 << 0) , **OVERTEMPERATURE\_INV** = (1 << 1) , **OVERVOLTAGE** = (1 << 2) ,  
**OVERCURRENT** = (1 << 3) , **OVERSPEED** = (1 << 4) , **UNDERVOLTAGE** = (1 << 5) , **CONTROL\_FAULT** = (1 << 6) ,  
**WARNING** = (1 << 7) , **OVERTEMPERATURE\_MOT** = (1 << 8) , **FEEDBACK\_FAULT** = (1 << 9) }  
*Enumeration of inverter error states.*

## Functions

- void `set_error` (volatile void \*data, `InverterError` error)  
*Sets an error in the error field of a data structure.*
- void `clear_error` (volatile void \*data, `InverterError` error)  
*Clears an error in the error field of a data structure.*
- bool `is_error_set` (volatile void \*data, `InverterError` error)  
*Checks if an error is set in the error field of a data structure.*

## 4.5.1 Macro Definition Documentation

### 4.5.1.1 OVERCURRENT\_TH

```
#define OVERCURRENT_TH 100.0f
```

[A] Threshold for instantaneous overcurrent fault

### 4.5.1.2 OVERSPEED\_TH

```
#define OVERSPEED_TH 20000.0f
```

[RPM] Threshold for motor overspeed fault

### 4.5.1.3 OVERTEMPERATURE\_INVERTER\_TH

```
#define OVERTEMPERATURE_INVERTER_TH 60.0f
```

[degC] Threshold for inverter overtemperature fault

### 4.5.1.4 OVERTEMPERATURE\_MOTOR\_TH

```
#define OVERTEMPERATURE_MOTOR_TH 90.0f
```

[degC] Threshold for motor overtemperature fault

### 4.5.1.5 OVERVOLTAGE\_TH

```
#define OVERVOLTAGE_TH 600.0f
```

[V] Threshold for overvoltage fault

### 4.5.1.6 UNDERVOLTAGE\_TH

```
#define UNDERVOLTAGE_TH 10.0f
```

[V] Threshold for undervoltage fault

## 4.5.2 Enumeration Type Documentation

### 4.5.2.1 InverterError

```
enum InverterError
```

Enumeration of inverter error states.

## Enumerator

NONE	
POWER_FAULT	Power fault error bit
OVERTEMPERATURE_INV	Inverter overtemperature error bit
OVERVOLTAGE	Overvoltage error bit
OVERCURRENT	Overcurrent error bit
OVERSPEED	Overspeed error bit
UNDERVOLTAGE	Undervoltage error bit
CONTROL_FAULT	Control fault error bit
WARNING	Warning error bit
OVERTEMPERATURE_MOT	Motor overtemperature error bit
FEEDBACK_FAULT	<a href="#">Feedback</a> fault error bit

### 4.5.3 Function Documentation

#### 4.5.3.1 clear\_error()

```
void clear_error (
    volatile void * data,
    InverterError error )
```

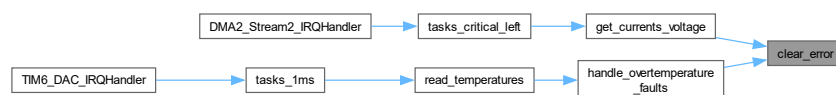
Clears an error in the error field of a data structure.

This function clears the specified error bit in the error field of a data structure.

#### Parameters

out	<i>data</i>	Pointer to the data structure containing the error field.
in	<i>error</i>	The error to be cleared. This should be one of the values from the InverterError enumeration.

Here is the caller graph for this function:



#### 4.5.3.2 is\_error\_set()

```
bool is_error_set (
    volatile void * data,
    InverterError error )
```

Checks if an error is set in the error field of a data structure.

This function checks if the specified error bit is set in the error field of a data structure.



## Parameters

in	<i>data</i>	Pointer to the data structure containing the error field.
in	<i>error</i>	The error to be checked. This should be one of the values from the InverterError enumeration.

## Returns

true if the specified error is set, false otherwise.

## 4.5.3.3 set\_error()

```
void set_error (
    volatile void * data,
    InverterError error )
```

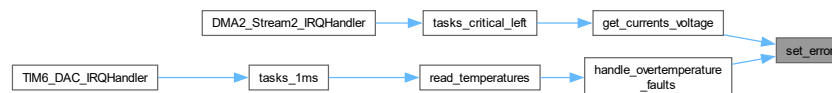
Sets an error in the error field of a data structure.

This function sets the specified error bit in the error field of a data structure.

## Parameters

out	<i>data</i>	Pointer to the data structure containing the error field.
in	<i>error</i>	The error to be set. This should be one of the values from the InverterError enumeration.

Here is the caller graph for this function:



## 4.6 ERRORS.h

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

```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019 #include <stdbool.h>
00020 #include <stdint.h>
00021
00025 typedef enum {
00026     NONE = 0,
00027     POWER_FAULT = (1 << 0),
00028     OVERTEMPERATURE_INV = (1 << 1),
00029     OVERVOLTAGE = (1 << 2),
00030     OVERCURRENT = (1 << 3),
00031     OVERSPEED = (1 << 4),
00032     UNDERVOLTAGE = (1 << 5),
00033     CONTROL_FAULT = (1 << 6),
00034     WARNING = (1 << 7),
00035     OVERTEMPERATURE_MOT = (1 << 8),
00036     FEEDBACK_FAULT = (1 << 9)
00037 } InverterError;
00038
```

```

00039 /* Define fault thresholds */
00040 #define OVERTEMPERATURE_INVERTER_TH 60.0f
00041 #define OVERVOLTAGE_TH 600.0f
00042 #define OVERCURRENT_TH 100.0f
00043 #define OVERSPEED_TH 20000.0f
00044 #define UNDERVOLTAGE_TH 10.0f
00045 #define OVERTEMPERATURE_MOTOR_TH 90.0f
00055 void set_error(volatile void *data, InverterError error);
00056
00065 void clear_error(volatile void *data, InverterError error);
00066
00076 bool is_error_set(volatile void *data, InverterError error);

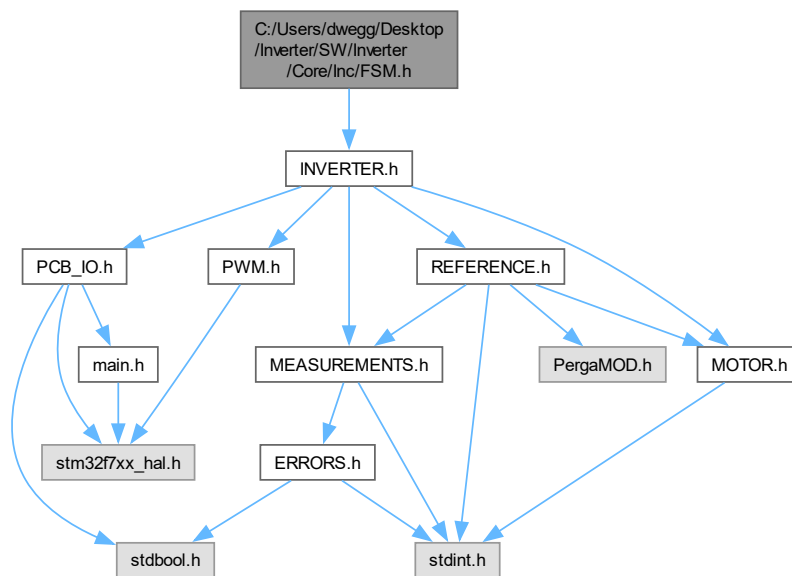
```

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

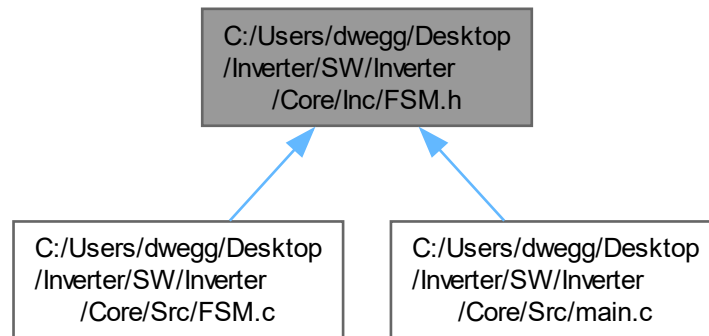
Header for the inverter Finite State Machine.

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

Include dependency graph for FSM.h:



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



## Functions

- void `eval_inv_FSM` (volatile `InverterStruct` \*inv)  
*Run the Finite State Machine (FSM) for inverter operation control.*

### 4.7.1 Detailed Description

Header for the inverter Finite State Machine.

#### 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>

### 4.7.2 Function Documentation

#### 4.7.2.1 `eval_inv_FSM()`

```
void eval_inv_FSM (  
    volatile 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:



## 4.8 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 #include "INVERTER.h" // inverter struct
00023
00024
00030 void eval_inv_FSM(volatile InverterStruct *inv);
00031
00032 #endif /* FSM_H */
  
```

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

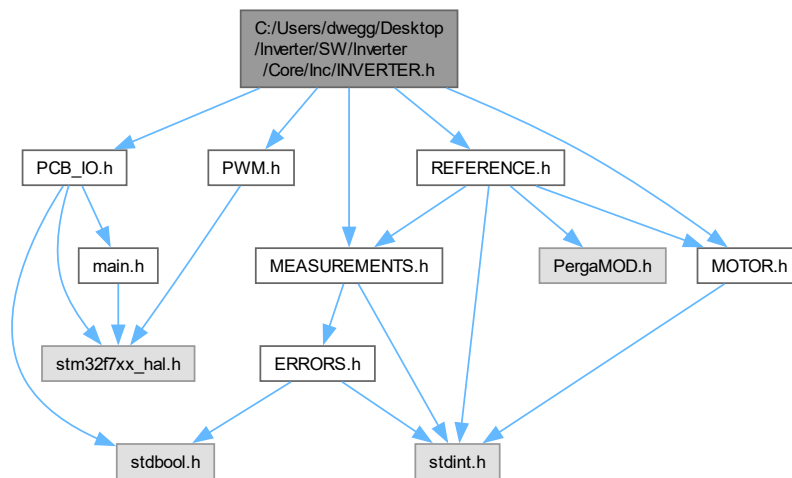
Header file for the inverter struct and extern variables.

```

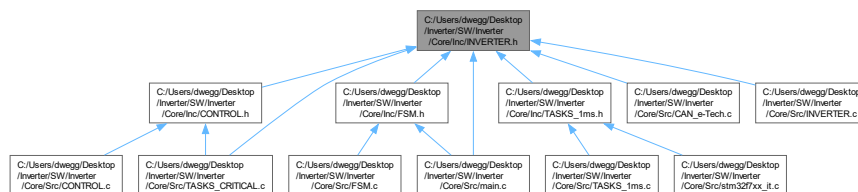
#include "PCB_IO.h"
#include "MEASUREMENTS.h"
#include "REFERENCE.h"
#include "MOTOR.h"
  
```

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

Include dependency graph for INVERTER.h:



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



## Data Structures

- struct [InverterStruct](#)  
*Inverter structure.*

## Macros

- #define [TS](#) 0.000025
- #define [DT](#) 0.00000015

## Enumerations

- enum [InverterState](#) { [INV\\_STATE\\_IDLE](#) , [INV\\_STATE\\_STARTUP](#) , [INV\\_STATE\\_RUNNING](#) , [INV\\_STATE\\_FAULT](#) }
- Enumeration of inverter operation states.*

## Functions

- void `initialize_inverter` (volatile `InverterStruct` \*inv, `LED` \*led, `GPIO_TypeDef` \*enable\_port, `uint16_t` enable\_pin, `TIM_HandleTypeDef` \*htim, `ADC_HandleTypeDef` \*hadc, `MotorParameters` \*motor, volatile `uint16_t` \*rawADC)  
*Initialize the inverter.*
- void `init_control_loops` (volatile `InverterStruct` \*inv, `MotorParameters` \*motor)  
*Initializes the id-iq current control PI controllers.*
- void `enable_control_loops` (volatile `InverterStruct` \*inv)  
*Enables the PI controllers.*
- void `disable_control_loops` (volatile `InverterStruct` \*inv)  
*Disables the PI controllers.*

## Variables

- volatile `InverterStruct` `inverter_left`  
*Left inverter structure.*
- volatile `InverterStruct` `inverter_right`  
*Right inverter structure.*

### 4.9.1 Detailed Description

Header file for the inverter struct and extern variables.

Attention

Copyright (c) 2024 David Redondo (@dweggg on 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>

### 4.9.2 Macro Definition Documentation

#### 4.9.2.1 DT

```
#define DT 0.00000015
```

Dead time in seconds (150 ns), time in which both top and bottom transistors are open

#### 4.9.2.2 TS

```
#define TS 0.000025
```

Switching time in seconds (25 us), inverse of the switching frequency of 40 kHz

### 4.9.3 Enumeration Type Documentation

#### 4.9.3.1 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

### 4.9.4 Function Documentation

#### 4.9.4.1 disable\_control\_loops()

```
void disable_control_loops (
    volatile InverterStruct * inv )
```

Disables the PI controllers.

##### Parameters

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

#### 4.9.4.2 enable\_control\_loops()

```
void enable_control_loops (
    volatile InverterStruct * inv )
```

Enables the PI controllers.

##### Parameters

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

#### 4.9.4.3 init\_control\_loops()

```
void init_control_loops (
    volatile InverterStruct * inv,
    MotorParameters * motor )
```

Initializes the id-iq current control PI controllers.

##### Parameters

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

Initializes the id-iq current control PI controllers.

## Parameters

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

Here is the caller graph for this function:



## 4.9.4.4 initialize\_inverter()

```

void initialize_inverter (
    volatile InverterStruct * inv,
    LED * led,
    GPIO_TypeDef * enable_port,
    uint16_t enable_pin,
    TIM_HandleTypeDef * htim,
    ADC_HandleTypeDef * hadc,
    MotorParameters * motor,
    volatile uint16_t * rawADC )
  
```

Initialize the inverter.

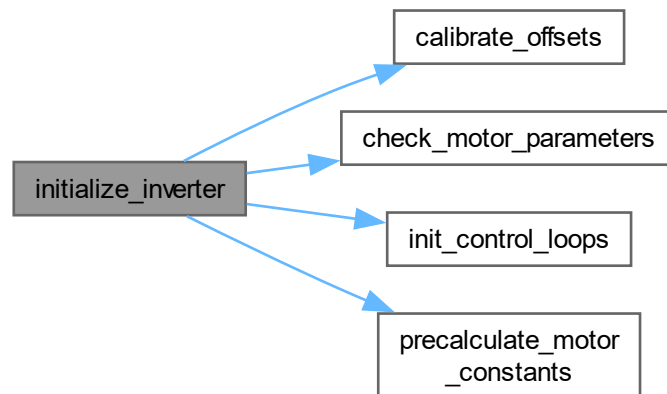
This function initializes the inverter structure with the specified [LED](#), GPIO port, and pin.

## Parameters

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



Here is the call graph for this function:



Here is the caller graph for this function:



## 4.9.5 Variable Documentation

### 4.9.5.1 inverter\_left

```
volatile InverterStruct inverter_left [extern]
```

Left inverter structure.

External declaration of the left inverter structure

External declaration of the left inverter structure.

### 4.9.5.2 inverter\_right

```
volatile InverterStruct inverter_right [extern]
```

Right inverter structure.

External declaration of the right inverter structure

External declaration of the right inverter structure.

## 4.10 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 "PCB_IO.h" // peripheral types
00024 #include "MEASUREMENTS.h" // a few structs
00025 #include "REFERENCE.h" // reference struct
00026 #include "MOTOR.h" // motor struct
00027 #include "PWM.h" // duties struct
00028
00029
00030 #define TS 0.000025
00031 #define DT 0.00000015
00037 typedef enum {
00038     INV_STATE_IDLE,
00039     INV_STATE_STARTUP,
00040     INV_STATE_RUNNING,
00041     INV_STATE_FAULT
00042 } InverterState;
00043
00047 typedef struct {
00048     LED *led;
00049     GPIO_TypeDef *enable_port;
00050     uint16_t enable_pin;
00051     TIM_HandleTypeDef *htim;
00052     ADC_HandleTypeDef *hadc;
00053     InverterState state;
00054     Analog analog;
00055     Encoder encoder;
00056     Feedback feedback;
00057     Reference reference;
00058     Duties duties;
00059     int8_t direction;
00060     float tempInverter;
00061     float tempMotor;
00062     MotorParameters *motor;
00063     pi_struct idLoop;
00064     pi_struct iqLoop;
00065     float vsMax;
00066     float vd;
00067     float vq;
00068     pi_struct speedLoop;
00069     InverterError errors;
00070     bool enable;
00071     bool enableSW;
00073 } InverterStruct;
00074
00075 extern volatile InverterStruct inverter_left;
00076 extern volatile InverterStruct inverter_right;
00091 void initialize_inverter(volatile InverterStruct *inv, LED *led, GPIO_TypeDef *enable_port, uint16_t
enable_pin, TIM_HandleTypeDef *htim, ADC_HandleTypeDef *hadc, MotorParameters *motor, volatile
uint16_t *rawADC);
00092
00098 void init_control_loops(volatile InverterStruct *inv, MotorParameters *motor);
00099
00105 void enable_control_loops(volatile InverterStruct *inv);
00106
00112 void disable_control_loops(volatile InverterStruct *inv);
00113
00114 #endif /* INVERTER_H */

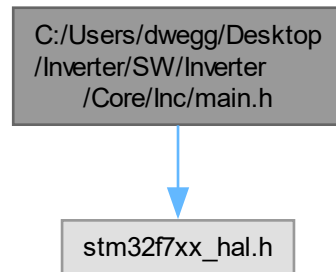
```

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

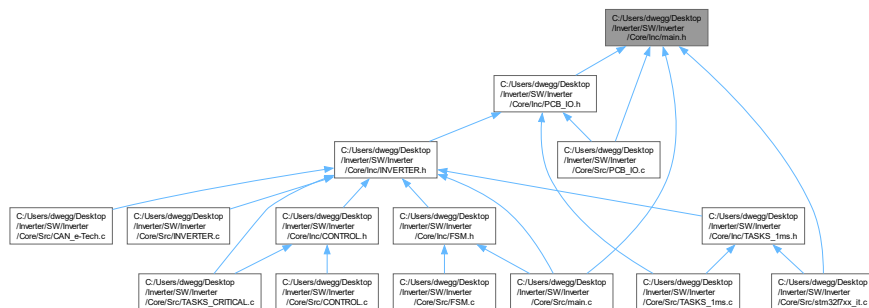
: Header for [main.c](#) file. This file contains the common defines of the application.

```
#include "stm32f7xx_hal.h"
```

Include dependency graph for main.h:



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



## Macros

- `#define 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.*

### 4.11.1 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.

### 4.11.2 Macro Definition Documentation

#### 4.11.2.1 A\_L\_GPIO\_Port

```
#define A_L_GPIO_Port GPIOD
```

#### 4.11.2.2 A\_L\_Pin

```
#define A_L_Pin GPIO_PIN_12
```

#### 4.11.2.3 A\_R\_GPIO\_Port

```
#define A_R_GPIO_Port GPIOA
```

#### 4.11.2.4 A\_R\_Pin

```
#define A_R_Pin GPIO_PIN_15
```

#### 4.11.2.5 B\_L\_GPIO\_Port

```
#define B_L_GPIO_Port GPIOD
```

#### 4.11.2.6 B\_L\_Pin

```
#define B_L_Pin GPIO_PIN_14
```

#### 4.11.2.7 B\_R\_GPIO\_Port

```
#define B_R_GPIO_Port GPIOB
```

#### 4.11.2.8 B\_R\_Pin

```
#define B_R_Pin GPIO_PIN_10
```

#### 4.11.2.9 DAC\_GPIO\_Port

```
#define DAC_GPIO_Port GPIOA
```

#### 4.11.2.10 DAC\_Pin

```
#define DAC_Pin GPIO_PIN_4
```

#### 4.11.2.11 DIR\_GPIO\_Port

```
#define DIR_GPIO_Port GPIOD
```

#### 4.11.2.12 DIR\_Pin

```
#define DIR_Pin GPIO_PIN_3
```

#### 4.11.2.13 ENABLE\_L\_GPIO\_Port

```
#define ENABLE_L_GPIO_Port GPIOE
```

#### 4.11.2.14 ENABLE\_L\_Pin

```
#define ENABLE_L_Pin GPIO_PIN_7
```

#### 4.11.2.15 ENABLE\_R\_GPIO\_Port

```
#define ENABLE_R_GPIO_Port GPIOB
```

#### 4.11.2.16 ENABLE\_R\_Pin

```
#define ENABLE_R_Pin GPIO_PIN_2
```

#### 4.11.2.17 ia\_L\_GPIO\_Port

```
#define ia_L_GPIO_Port GPIOA
```

#### 4.11.2.18 ia\_L\_Pin

```
#define ia_L_Pin GPIO_PIN_0
```

#### 4.11.2.19 ia\_R\_GPIO\_Port

```
#define ia_R_GPIO_Port GPIOA
```

#### 4.11.2.20 ia\_R\_Pin

```
#define ia_R_Pin GPIO_PIN_6
```

#### 4.11.2.21 ib\_L\_GPIO\_Port

```
#define ib_L_GPIO_Port GPIOA
```

#### 4.11.2.22 ib\_L\_Pin

```
#define ib_L_Pin GPIO_PIN_1
```

#### 4.11.2.23 ib\_R\_GPIO\_Port

```
#define ib_R_GPIO_Port GPIOA
```

#### 4.11.2.24 ib\_R\_Pin

```
#define ib_R_Pin GPIO_PIN_7
```

#### 4.11.2.25 ic\_L\_GPIO\_Port

```
#define ic_L_GPIO_Port GPIOA
```

#### 4.11.2.26 ic\_L\_Pin

```
#define ic_L_Pin GPIO_PIN_2
```

#### 4.11.2.27 ic\_R\_GPIO\_Port

```
#define ic_R_GPIO_Port GPIOB
```

#### 4.11.2.28 ic\_R\_Pin

```
#define ic_R_Pin GPIO_PIN_0
```

#### 4.11.2.29 LED\_ERR\_GPIO\_Port

```
#define LED_ERR_GPIO_Port GPIOD
```

#### 4.11.2.30 LED\_ERR\_Pin

```
#define LED_ERR_Pin GPIO_PIN_6
```

#### 4.11.2.31 LED\_LEFT\_GPIO\_Port

```
#define LED_LEFT_GPIO_Port GPIOD
```

#### 4.11.2.32 LED\_LEFT\_Pin

```
#define LED_LEFT_Pin GPIO_PIN_4
```

#### 4.11.2.33 LED\_RIGHT\_GPIO\_Port

```
#define LED_RIGHT_GPIO_Port GPIOD
```



#### 4.11.2.34 LED\_RIGHT\_Pin

```
#define LED_RIGHT_Pin GPIO_PIN_5
```

#### 4.11.2.35 PWM1\_L\_GPIO\_Port

```
#define PWM1_L_GPIO_Port GPIOE
```

#### 4.11.2.36 PWM1\_L\_Pin

```
#define PWM1_L_Pin GPIO_PIN_8
```

#### 4.11.2.37 PWM1\_R\_GPIO\_Port

```
#define PWM1_R_GPIO_Port GPIOA
```

#### 4.11.2.38 PWM1\_R\_Pin

```
#define PWM1_R_Pin GPIO_PIN_5
```

#### 4.11.2.39 PWM2\_L\_GPIO\_Port

```
#define PWM2_L_GPIO_Port GPIOE
```

#### 4.11.2.40 PWM2\_L\_Pin

```
#define PWM2_L_Pin GPIO_PIN_9
```

#### 4.11.2.41 PWM2\_R\_GPIO\_Port

```
#define PWM2_R_GPIO_Port GPIOC
```

#### 4.11.2.42 PWM2\_R\_Pin

```
#define PWM2_R_Pin GPIO_PIN_6
```

#### 4.11.2.43 PWM3\_L\_GPIO\_Port

```
#define PWM3_L_GPIO_Port GPIOE
```

#### 4.11.2.44 PWM3\_L\_Pin

```
#define PWM3_L_Pin GPIO_PIN_10
```

#### 4.11.2.45 PWM3\_R\_GPIO\_Port

```
#define PWM3_R_GPIO_Port GPIOB
```

#### 4.11.2.46 PWM3\_R\_Pin

```
#define PWM3_R_Pin GPIO_PIN_14
```

#### 4.11.2.47 PWM4\_L\_GPIO\_Port

```
#define PWM4_L_GPIO_Port GPIOE
```

#### 4.11.2.48 PWM4\_L\_Pin

```
#define PWM4_L_Pin GPIO_PIN_11
```

#### 4.11.2.49 PWM4\_R\_GPIO\_Port

```
#define PWM4_R_GPIO_Port GPIOC
```

#### 4.11.2.50 PWM4\_R\_Pin

```
#define PWM4_R_Pin GPIO_PIN_7
```

#### 4.11.2.51 PWM5\_L\_GPIO\_Port

```
#define PWM5_L_GPIO_Port GPIOE
```

#### 4.11.2.52 PWM5\_L\_Pin

```
#define PWM5_L_Pin GPIO_PIN_12
```

#### 4.11.2.53 PWM5\_R\_GPIO\_Port

```
#define PWM5_R_GPIO_Port GPIOB
```

#### 4.11.2.54 PWM5\_R\_Pin

```
#define PWM5_R_Pin GPIO_PIN_15
```

#### 4.11.2.55 PWM6\_L\_GPIO\_Port

```
#define PWM6_L_GPIO_Port GPIOE
```

#### 4.11.2.56 PWM6\_L\_Pin

```
#define PWM6_L_Pin GPIO_PIN_13
```

#### 4.11.2.57 PWM6\_R\_GPIO\_Port

```
#define PWM6_R_GPIO_Port GPIOC
```

#### 4.11.2.58 PWM6\_R\_Pin

```
#define PWM6_R_Pin GPIO_PIN_8
```

#### 4.11.2.59 SC\_det\_GPIO\_Port

```
#define SC_det_GPIO_Port GPIOC
```

#### 4.11.2.60 SC\_det\_Pin

```
#define SC_det_Pin GPIO_PIN_4
```

#### 4.11.2.61 Tinv\_L\_GPIO\_Port

```
#define Tinv_L_GPIO_Port GPIOC
```

#### 4.11.2.62 Tinv\_L\_Pin

```
#define Tinv_L_Pin GPIO_PIN_0
```

#### 4.11.2.63 Tinv\_R\_GPIO\_Port

```
#define Tinv_R_GPIO_Port GPIOC
```

#### 4.11.2.64 Tinv\_R\_Pin

```
#define Tinv_R_Pin GPIO_PIN_1
```

#### 4.11.2.65 Tmot\_L\_GPIO\_Port

```
#define Tmot_L_GPIO_Port GPIOC
```

#### 4.11.2.66 Tmot\_L\_Pin

```
#define Tmot_L_Pin GPIO_PIN_2
```

#### 4.11.2.67 Tmot\_R\_GPIO\_Port

```
#define Tmot_R_GPIO_Port GPIOC
```

#### 4.11.2.68 Tmot\_R\_Pin

```
#define Tmot_R_Pin GPIO_PIN_3
```

#### 4.11.2.69 TRIP\_L\_GPIO\_Port

```
#define TRIP_L_GPIO_Port GPIOA
```

#### 4.11.2.70 TRIP\_L\_Pin

```
#define TRIP_L_Pin GPIO_PIN_8
```

#### 4.11.2.71 TRIP\_R\_GPIO\_Port

```
#define TRIP_R_GPIO_Port GPIOC
```

#### 4.11.2.72 TRIP\_R\_Pin

```
#define TRIP_R_Pin GPIO_PIN_9
```

#### 4.11.2.73 VDC\_L\_GPIO\_Port

```
#define VDC_L_GPIO_Port GPIOA
```

#### 4.11.2.74 VDC\_L\_Pin

```
#define VDC_L_Pin GPIO_PIN_3
```

#### 4.11.2.75 VDC\_R\_GPIO\_Port

```
#define VDC_R_GPIO_Port GPIOB
```

#### 4.11.2.76 VDC\_R\_Pin

```
#define VDC_R_Pin GPIO_PIN_1
```

#### 4.11.2.77 WRN\_L\_GPIO\_Port

```
#define WRN_L_GPIO_Port GPIOE
```

#### 4.11.2.78 WRN\_L\_Pin

```
#define WRN_L_Pin GPIO_PIN_14
```

#### 4.11.2.79 WRN\_R\_GPIO\_Port

```
#define WRN_R_GPIO_Port GPIOE
```

#### 4.11.2.80 WRN\_R\_Pin

```
#define WRN_R_Pin GPIO_PIN_15
```

#### 4.11.2.81 Z\_L\_GPIO\_Port

```
#define Z_L_GPIO_Port GPIOD
```

#### 4.11.2.82 Z\_L\_Pin

```
#define Z_L_Pin GPIO_PIN_15
```

#### 4.11.2.83 Z\_R\_GPIO\_Port

```
#define Z_R_GPIO_Port GPIOB
```

#### 4.11.2.84 Z\_R\_Pin

```
#define Z_R_Pin GPIO_PIN_11
```

### 4.11.3 Function Documentation

#### 4.11.3.1 Error\_Handler()

```
void Error_Handler (  
    void )
```

This function is executed in case of error occurrence.

## Return values

None	
------	--

Here is the caller graph for this function:



## 4.12 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 Tinv_L_Pin GPIO_PIN_0
00061 #define Tinv_L_GPIO_Port GPIOC
00062 #define Tinv_R_Pin GPIO_PIN_1
00063 #define Tinv_R_GPIO_Port GPIOC
00064 #define Tmot_L_Pin GPIO_PIN_2
00065 #define Tmot_L_GPIO_Port GPIOC
00066 #define Tmot_R_Pin GPIO_PIN_3
00067 #define Tmot_R_GPIO_Port GPIOC
00068 #define ia_L_Pin GPIO_PIN_0
00069 #define ia_L_GPIO_Port GPIOA
00070 #define ib_L_Pin GPIO_PIN_1

```

```
00071 #define ib_L_GPIO_Port GPIOA
00072 #define ic_L_Pin GPIO_PIN_2
00073 #define ic_L_GPIO_Port GPIOA
00074 #define VDC_L_Pin GPIO_PIN_3
00075 #define VDC_L_GPIO_Port GPIOA
00076 #define DAC_Pin GPIO_PIN_4
00077 #define DAC_GPIO_Port GPIOA
00078 #define PWM1_R_Pin GPIO_PIN_5
00079 #define PWM1_R_GPIO_Port GPIOA
00080 #define ia_R_Pin GPIO_PIN_6
00081 #define ia_R_GPIO_Port GPIOA
00082 #define ib_R_Pin GPIO_PIN_7
00083 #define ib_R_GPIO_Port GPIOA
00084 #define SC_det_Pin GPIO_PIN_4
00085 #define SC_det_GPIO_Port GPIOC
00086 #define ic_R_Pin GPIO_PIN_0
00087 #define ic_R_GPIO_Port GPIOB
00088 #define VDC_R_Pin GPIO_PIN_1
00089 #define VDC_R_GPIO_Port GPIOB
00090 #define ENABLE_R_Pin GPIO_PIN_2
00091 #define ENABLE_R_GPIO_Port GPIOB
00092 #define ENABLE_L_Pin GPIO_PIN_7
00093 #define ENABLE_L_GPIO_Port GPIOE
00094 #define PWM1_L_Pin GPIO_PIN_8
00095 #define PWM1_L_GPIO_Port GPIOE
00096 #define PWM2_L_Pin GPIO_PIN_9
00097 #define PWM2_L_GPIO_Port GPIOE
00098 #define PWM3_L_Pin GPIO_PIN_10
00099 #define PWM3_L_GPIO_Port GPIOE
00100 #define PWM4_L_Pin GPIO_PIN_11
00101 #define PWM4_L_GPIO_Port GPIOE
00102 #define PWM5_L_Pin GPIO_PIN_12
00103 #define PWM5_L_GPIO_Port GPIOE
00104 #define PWM6_L_Pin GPIO_PIN_13
00105 #define PWM6_L_GPIO_Port GPIOE
00106 #define WRN_L_Pin GPIO_PIN_14
00107 #define WRN_L_GPIO_Port GPIOE
00108 #define WRN_R_Pin GPIO_PIN_15
00109 #define WRN_R_GPIO_Port GPIOE
00110 #define B_R_Pin GPIO_PIN_10
00111 #define B_R_GPIO_Port GPIOB
00112 #define Z_R_Pin GPIO_PIN_11
00113 #define Z_R_GPIO_Port GPIOB
00114 #define PWM3_R_Pin GPIO_PIN_14
00115 #define PWM3_R_GPIO_Port GPIOB
00116 #define PWM5_R_Pin GPIO_PIN_15
00117 #define PWM5_R_GPIO_Port GPIOB
00118 #define A_L_Pin GPIO_PIN_12
00119 #define A_L_GPIO_Port GPIOD
00120 #define B_L_Pin GPIO_PIN_14
00121 #define B_L_GPIO_Port GPIOD
00122 #define Z_L_Pin GPIO_PIN_15
00123 #define Z_L_GPIO_Port GPIOD
00124 #define PWM2_R_Pin GPIO_PIN_6
00125 #define PWM2_R_GPIO_Port GPIOC
00126 #define PWM4_R_Pin GPIO_PIN_7
00127 #define PWM4_R_GPIO_Port GPIOC
00128 #define PWM6_R_Pin GPIO_PIN_8
00129 #define PWM6_R_GPIO_Port GPIOC
00130 #define TRIP_R_Pin GPIO_PIN_9
00131 #define TRIP_R_GPIO_Port GPIOC
00132 #define TRIP_L_Pin GPIO_PIN_8
00133 #define TRIP_L_GPIO_Port GPIOA
00134 #define A_R_Pin GPIO_PIN_15
00135 #define A_R_GPIO_Port GPIOA
00136 #define DIR_Pin GPIO_PIN_3
00137 #define DIR_GPIO_Port GPIOD
00138 #define LED_LEFT_Pin GPIO_PIN_4
00139 #define LED_LEFT_GPIO_Port GPIOD
00140 #define LED_RIGHT_Pin GPIO_PIN_5
00141 #define LED_RIGHT_GPIO_Port GPIOD
00142 #define LED_ERR_Pin GPIO_PIN_6
00143 #define LED_ERR_GPIO_Port GPIOD
00144
00145 /* USER CODE BEGIN Private defines */
00146
00147 /* USER CODE END Private defines */
00148
00149 #ifndef __cplusplus
00150 }
00151 #endif
00152
00153 #endif /* __MAIN_H */
```





## Macros

- #define `CURRENT_SLOPE` 117.57704f
- #define `CURRENT_OFFSET` 1.70068027211f
- #define `VOLTAGE_SLOPE` 263.435f
- #define `VOLTAGE_OFFSET` 0.02083f

## Functions

- uint8\_t `get_currents_voltage` (volatile uint16\_t ADC\_raw[], volatile Analog \*analog, volatile Feedback \*feedback, volatile InverterError \*errors, float sinTheta\_e, float cosTheta\_e)  
*Get electrical ADC measurements.*
- float `get_linear` (uint32\_t bits, float slope, float offset)  
*Convert ADC reading to physical measurement with linear response.*
- void `get_idiq` (float ia, float ib, float ic, float sinTheta\_e, float cosTheta\_e, float \*idMeas, float \*iqMeas)  
*Computes d-q currents from current measurements and electrical angle.*
- float `get_temperature` (uint32\_t bits, const float tempLUT[])  
*Retrieves temperature from a lookup table based on ADC bits.*
- void `calibrate_offsets` (volatile uint16\_t rawADC[], volatile float currentOffsets[], uint32\_t numSamples)  
*Calibrate the current sensor offsets.*

## Variables

- const float `tempInverterLUT` []
- const float `tempMotorLUT` []
- volatile uint16\_t `rawADC_left` [4]  
*Raw ADC data for the left inverter.*
- volatile uint16\_t `rawADC_right` [4]  
*Raw ADC data for the right inverter.*
- volatile uint16\_t `rawADC_temp` [4]  
*Raw ADC data for the temperatures.*

### 4.13.1 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>

## 4.13.2 Macro Definition Documentation

### 4.13.2.1 CURRENT\_OFFSET

```
#define CURRENT_OFFSET 1.70068027211f
```

[V]  $(10/(4.7+10)) * 2.5 \text{ V}$  (not actually used, self calibration at start)

### 4.13.2.2 CURRENT\_SLOPE

```
#define CURRENT_SLOPE 117.57704f
```

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

### 4.13.2.3 VOLTAGE\_OFFSET

```
#define VOLTAGE_OFFSET 0.02083f
```

[V]  $(100/(4700+100)) * 5 \text{ V}$

### 4.13.2.4 VOLTAGE\_SLOPE

```
#define VOLTAGE_SLOPE 263.435f
```

[V/V]  $1/(1/3 * 0.011388) \text{ V}$

## 4.13.3 Function Documentation

### 4.13.3.1 calibrate\_offsets()

```
void calibrate_offsets (
    volatile uint16_t rawADC[],
    volatile float currentOffsets[],
    uint32_t numSamples )
```

Calibrate the current sensor offsets.

This function calculates the average offset for each current sensor channel by reading the ADC values when no current is flowing. The calculated offsets are used to correct the sensor readings.

#### Parameters

in	<i>rawADC</i>	Buffer containing the raw ADC values for the channels.
out	<i>currentOffsets</i>	Array to store the calculated offsets for each current channel.
in	<i>numSamples</i>	Number of samples to average for the offset calculation.

Here is the caller graph for this function:



#### 4.13.3.2 get\_currents\_voltage()

```

uint8_t get_currents_voltage (
    volatile uint16_t ADC_raw[],
    volatile Analog * analog,
    volatile Feedback * feedback,
    volatile InverterError * errors,
    float sinTheta_e,
    float cosTheta_e )
  
```

Get electrical ADC measurements.

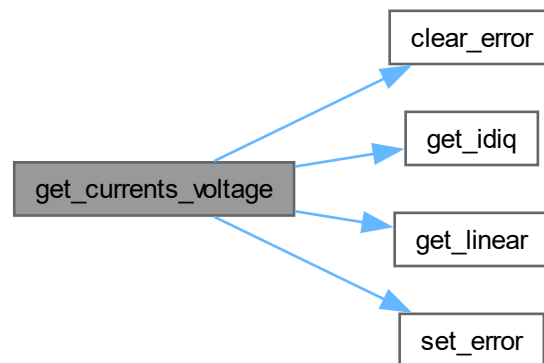
##### Parameters

in	<i>ADC_raw</i>	Pointer to the raw ADC values array.
out	<i>analog</i>	Pointer to the ADC struct to store the results.
out	<i>feedback</i>	Pointer to the <a href="#">Feedback</a> struct to store id and iq.
in	<i>sinTheta_e</i>	Electrical angle sine (-1..1)
in	<i>cosTheta_e</i>	Electrical angle cosine (-1..1)

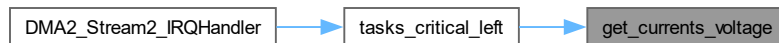
##### Return values

<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:



#### 4.13.3.3 get\_idiq()

```

void get_idiq (
    float ia,
    float ib,
    float ic,
    float sinTheta_e,
    float cosTheta_e,
    float * idMeas,
    float * iqMeas )
  
```

Computes d-q currents from current measurements and electrical angle.

This function computes the d-q currents from phase currents (ABC), `theta_e`, and stores the results in the provided pointers.

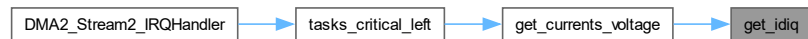
##### Parameters

in	<i>ia</i>	Phase A current in A.
in	<i>ib</i>	Phase B current in A.
in	<i>ic</i>	Phase C current in A.
in	<i>sinTheta_e</i> <i>_e</i>	Electrical angle sine (-1..1)

**Parameters**

in	<i>cosTheta_e</i>	Electrical angle cosine (-1..1)
out	<i>idMeas</i>	Pointer to store the D-axis current.
out	<i>iqMeas</i>	Pointer to store the Q-axis current.

Here is the caller graph for this function:

**4.13.3.4 get\_linear()**

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

Convert ADC reading to physical measurement with linear response.

**Parameters**

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

**Return values**

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

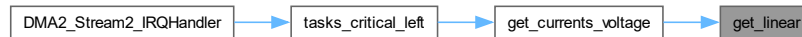
**Parameters**

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

**Return values**

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

Here is the caller graph for this function:



#### 4.13.3.5 get\_temperature()

```
float get_temperature (
    uint32_t bits,
    const float tempLUT[] )
```

Retrieves temperature from a lookup table based on ADC bits.

This function retrieves temperature from a lookup table based on the ADC bits. The lookup table (LUT) must have a value for each possible ADC bit combination.

##### Parameters

in	<i>bits</i>	ADC reading converted to bits.
in	<i>tempLUT</i>	Lookup table containing temperature values.

##### Returns

Temperature corresponding to the provided ADC bits.

Here is the caller graph for this function:



### 4.13.4 Variable Documentation

#### 4.13.4.1 rawADC\_left

```
volatile uint16_t rawADC_left[4] [extern]
```

Raw ADC data for the left inverter.

External declaration of raw ADC data for the left inverter

External declaration of raw ADC data for the left inverter.

**4.13.4.2 rawADC\_right**

```
volatile uint16_t rawADC_right[4] [extern]
```

Raw ADC data for the right inverter.

External declaration of raw ADC data for the right inverter

External declaration of raw ADC data for the right inverter.

**4.13.4.3 rawADC\_temp**

```
volatile uint16_t rawADC_temp[4] [extern]
```

Raw ADC data for the temperatures.

External declaration of raw ADC data for the temperatures

External declaration of raw ADC data for the temperature readings.

**4.13.4.4 tempInverterLUT**

```
const float tempInverterLUT[] [extern]
```

**4.13.4.5 tempMotorLUT**

```
const float tempMotorLUT[] [extern]
```

**4.14 MEASUREMENTS.h**

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

```
00001 /* USER CODE BEGIN Header */
00017 /* USER CODE END Header */
00018
00019
00020 #ifndef MEASUREMENTS_H
00021 #define MEASUREMENTS_H
00022
00023 #include <stdint.h>
00024 #include "ERRORS.h"
00025
00026 /* Define current and voltage gains/offsets */
00027 #define CURRENT_SLOPE 117.57704f
00028 #define CURRENT_OFFSET 1.70068027211f
00030 #define VOLTAGE_SLOPE 263.435f
00031 #define VOLTAGE_OFFSET 0.02083f
00033 extern const float tempInverterLUT[];
00034 extern const float tempMotorLUT[];
00035
00036 extern volatile uint16_t rawADC_left[4];
00037 extern volatile uint16_t rawADC_right[4];
00038 extern volatile uint16_t rawADC_temp[4];
00044 typedef struct {
00045     uint16_t A;
00046     uint16_t B;
00047     uint16_t Z;
00048     float we;
00049     float theta_e;
00050     float sinTheta_e;
00051     float cosTheta_e;
```

```

00052     uint8_t directionMeas;
00053 } Encoder;
00054
00055
00059 typedef struct {
00060     float ia;
00061     float ib;
00062     float ic;
00063     float vDC;
00064     float currentOffsets[3];
00065 } Analog;
00066
00067
00071 typedef struct {
00072     float idMeas;
00073     float iqMeas;
00074     float torqueCalc;
00075     float speedMeas;
00076 } Feedback;
00077
00078
00079
00089 uint8_t get_currents_voltage(volatile uint16_t ADC_raw[], volatile Analog* analog, volatile Feedback*
    feedback, volatile InverterError *errors, float sinTheta_e, float cosTheta_e);
00090
00098 float get_linear(uint32_t bits, float slope, float offset);
00099
00100
00115 void get_idiq(float ia, float ib, float ic, float sinTheta_e, float cosTheta_e, float *idMeas, float
    *iqMeas);
00116
00117
00128 float get_temperature(uint32_t bits, const float tempLUT[]);
00129
00130
00142 void calibrate_offsets(volatile uint16_t rawADC[], volatile float currentOffsets[], uint32_t
    numSamples);
00143
00144
00145 #endif /* MEASUREMENTS_H */

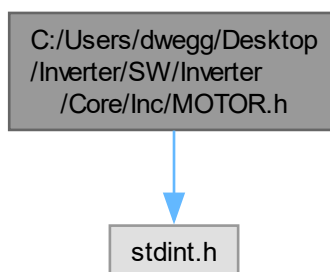
```

## 4.15 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/MOTOR.h File Reference

Header file for motor parameters.

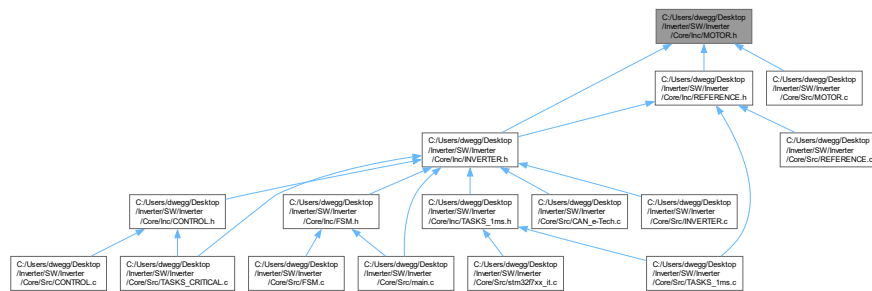
```
#include <stdint.h>
```

Include dependency graph for MOTOR.h:





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



## Data Structures

- struct [MotorConstants](#)  
Structure to hold precomputed motor constants.
- struct [MotorParameters](#)  
Structure to hold motor parameters.

## Functions

- void [precalculate\\_motor\\_constants](#) ([MotorParameters](#) \*motor)  
Precomputes the constants for a motor and updates the [MotorParameters](#) structure.
- int [check\\_motor\\_parameters](#) ([MotorParameters](#) \*motor, float Ts)  
Perform a parameter check and correct possible errors.

## Variables

- [MotorParameters motor\\_left](#)  
Left motor parameters.
- [MotorParameters motor\\_right](#)  
Right motor parameters.

### 4.15.1 Detailed Description

Header file for motor parameters.

#### Attention

Copyright (c) 2024 David Redondo (@dwegg on 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>

## 4.15.2 Function Documentation

### 4.15.2.1 `check_motor_parameters()`

```
int check_motor_parameters (
    MotorParameters * motor,
    float Ts )
```

Perform a parameter check and correct possible errors.

**Parameters**

<i>in</i>	<i>motor</i>	Pointer to the <a href="#">MotorParameters</a> struct.
-----------	--------------	--

**Return values**

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

Here is the caller graph for this function:

**4.15.2.2 precalculate\_motor\_constants()**

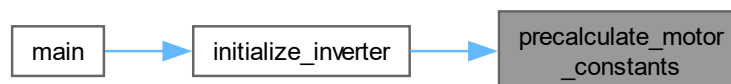
```
void precalculate_motor_constants (
    MotorParameters * motor )
```

Precomputes the constants for a motor and updates the [MotorParameters](#) structure.

**Parameters**

<i>motor</i>	[in, out] Pointer to the motor parameters structure
--------------	---

Here is the caller graph for this function:

**4.15.3 Variable Documentation****4.15.3.1 motor\_left**

```
MotorParameters motor_left [extern]
```

Left motor parameters.

#### 4.15.3.2 motor\_right

`MotorParameters` motor\_right [extern]

Right motor parameters.

### 4.16 MOTOR.h

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

```
00001 /* USER CODE BEGIN Header */
00017 /* USER CODE END Header */
00018
00019 #ifndef MOTOR_H
00020 #define MOTOR_H
00021
00022 #include <stdint.h>
00023
00027 typedef struct {
00028     float threePpLambda;
00029     float threePpLdMinusLq;
00030     float invThreePpLambda;
00031     float isc;
00032     float torqueBase;
00033     float invTorqueBase;
00034     float xi;
00035     float xiSquared;
00036     float oneMinusXi;
00037     float twoMinusXi;
00038     float fourTimesOneMinusXi;
00039     float eightTimesOneMinusXiSquared;
00040     float twoMinusXiSquared;
00041     float twoTimesOneMinusXiOnePlusXiSquared;
00042     float twoTimesOneMinusXiXiSquared;
00043     float fourTimesOneMinusXiOnePlusXiSquared;
00044     float fourTimesOneMinusXiXiSquared;
00045     float lambdaDivLqMinusLd;
00046     float betaMinusIsc;
00047 } MotorConstants;
00048
00052 typedef struct {
00053     float Ld;
00054     float Lq;
00055     float Rs;
00056     float lambda;
00057     uint8_t pp;
00058     float J;
00059     float b;
00060     float torqueMax;
00061     float dTorqueMax;
00062     float speedMax_RPM;
00063     float iMax;
00064     float vDCMax;
00065     MotorConstants constants;
00066 } MotorParameters;
00067
00068 extern MotorParameters motor_left;
00069 extern MotorParameters motor_right;
00070
00076 void precalculate_motor_constants(MotorParameters* motor);
00077
00083 int check_motor_parameters(MotorParameters *motor, float Ts);
00084 #endif /* MOTOR_H */
```

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

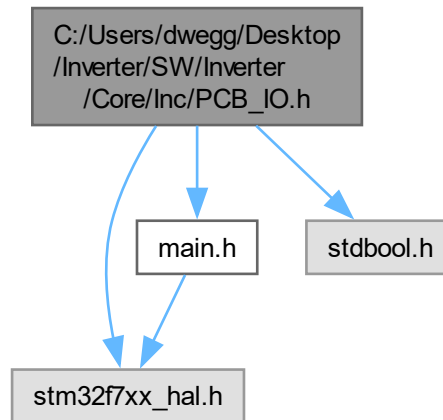
#### File Reference

Header file for handling GPIOs.

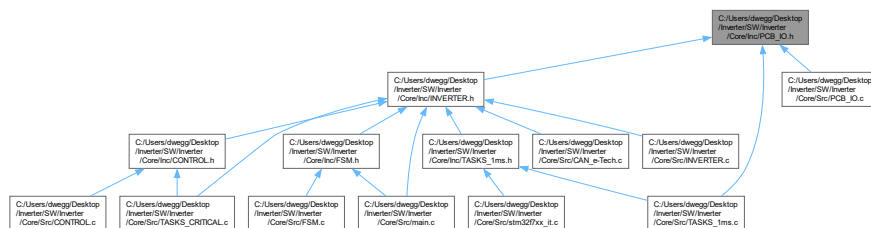
```
#include "stm32f7xx_hal.h"
#include "main.h"
```

```
#include <stdbool.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 `handle_LED` (`LED *led`, `uint32_t ms_counter`)  
*LED handler function.*
- void `handle_direction` (`volatile int8_t *dir_left`, `volatile int8_t *dir_right`)  
*Handles the direction of the motors.*
- void `enable_inverters` (`volatile bool enableSW_left`, `volatile bool enableSW_right`, `volatile bool *enable_left`, `volatile bool *enable_right`)  
*Handles the direction of the motors and enables/disables the inverters.*

## Variables

- `LED led_left`
- `LED led_right`
- `LED ledError`

### 4.17.1 Detailed Description

Header file for handling GPIOs.

#### 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>

### 4.17.2 Macro Definition Documentation

#### 4.17.2.1 DIR\_STATE

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

#### 4.17.2.2 DISABLE

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

#### 4.17.2.3 ENABLE

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

#### 4.17.2.4 SC\_DET\_STATE

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

#### 4.17.2.5 WRN\_STATE

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

### 4.17.3 Enumeration Type Documentation

#### 4.17.3.1 LEDMode

```
enum LEDMode
```

Enumerator

LED_MODE_BLINK_FAST	Fast blink mode
LED_MODE_BLINK_SLOW	Slow blink mode
LED_MODE_ON	LED on mode
LED_MODE_OFF	LED off mode

### 4.17.4 Function Documentation

#### 4.17.4.1 enable\_inverters()

```
void enable_inverters (  
    volatile bool enableSW_left,  
    volatile bool enableSW_right,  
    volatile bool * enable_left,  
    volatile bool * enable_right )
```

Handles the direction of the motors and enables/disables the inverters.

This function reads the state of the shutdown chain (SC or SDC) and enables/disables the inverters based on that and an external software enable bool.

Parameters

in	<i>enableSW_left</i>	The software enable state for the left inverter.
in	<i>enableSW_right</i>	The software enable state for the right inverter.
out	<i>enable_left</i>	Output parameter for the left inverter's enable state.
out	<i>enable_right</i>	Output parameter for the right inverter's enable state.

Here is the caller graph for this function:



#### 4.17.4.2 handle\_direction()

```

void handle_direction (
    volatile int8_t * dir_left,
    volatile int8_t * dir_right )
  
```

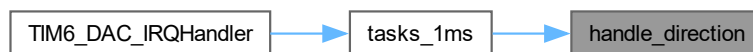
Handles the direction of the motors.

This function reads the state of the DIR switch and updates the directions of both the left and right motors. If one motor is set to rotate clockwise (CW), the other one is set to rotate counterclockwise (CCW), and vice versa.

##### Parameters

<i>dir_left</i>	Pointer to the direction parameter in the left inverter structure.
<i>dir_right</i>	Pointer to the direction parameter in the right inverter structure.

Here is the caller graph for this function:



#### 4.17.4.3 handle\_LED()

```

void handle_LED (
    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:



## 4.17.5 Variable Documentation

### 4.17.5.1 led\_left

```
LED led_left [extern]
```

### 4.17.5.2 led\_right

```
LED led_right [extern]
```

### 4.17.5.3 ledError

```
LED ledError [extern]
```

## 4.18 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 #include "main.h" // pin names/ports
00026 #include <stdbool.h>
00027
00028 // Read SC_det and DIR GPIOs
00029 #define SC_DET_STATE() (HAL_GPIO_ReadPin(SC_det_GPIO_Port, SC_det_Pin))
00030 #define DIR_STATE() (HAL_GPIO_ReadPin(DIR_GPIO_Port, DIR_Pin))
00031
00032 // Read WRN GPIOs
00033 #define WRN_STATE(port, pin) (HAL_GPIO_ReadPin(port, pin))
00034
00035 // Control ENABLE GPIOs
00036 #define ENABLE(port, pin) do { HAL_GPIO_WritePin(port, pin, GPIO_PIN_SET); } while(0)
  
```

```

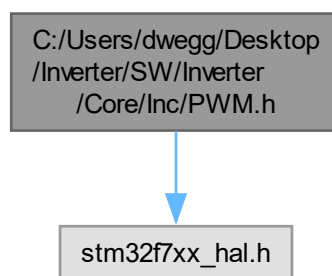
00037 #define DISABLE(port, pin)          do { HAL_GPIO_WritePin(port, pin, GPIO_PIN_RESET); } while(0)
00038
00039 // Define LED modes
00040 typedef enum {
00041     LED_MODE_BLINK_FAST,
00042     LED_MODE_BLINK_SLOW,
00043     LED_MODE_ON,
00044     LED_MODE_OFF
00045 } LEDMode;
00046
00047 typedef struct {
00048     GPIO_TypeDef *port;
00049     uint16_t pin;
00050     LEDMode mode;
00051 } LED;
00052
00053 // Declare LED variables as extern
00054 extern LED led_left;
00055 extern LED led_right;
00056 extern LED ledError;
00057
00058 // Function prototypes
00059 void handle_LED(LED *led, uint32_t ms_counter);
00060
00061 void handle_direction(volatile int8_t *dir_left, volatile int8_t *dir_right);
00062
00063 void enable_inverters(volatile bool enableSW_left, volatile bool enableSW_right, volatile bool
    *enable_left, volatile bool *enable_right);
00064
00065 #endif /* PCB_IO_H */

```

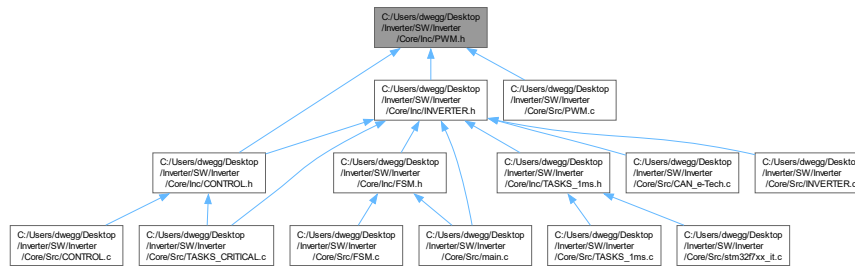
## 4.19 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PWM.h File Reference

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.*

## Functions

- void [enable\\_PWM](#) (TIM\_HandleTypeDef \*htim)  
*Enable PWM output.*
- void [disable\\_PWM](#) (TIM\_HandleTypeDef \*htim)  
*Disable PWM output.*
- void [update\\_PWM](#) (TIM\_HandleTypeDef \*htim, [Duties](#) duties)  
*Set PWM duty cycles.*

### 4.19.1 Detailed Description

Header file for controlling PWM output.

#### Attention

Copyright (c) 2024 David Redondo (@dwegg 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>

### 4.19.2 Function Documentation

#### 4.19.2.1 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.
-------------	---

**4.19.2.2 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.
-------------	---

**4.19.2.3 update\_PWM()**

```
void update_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>	<a href="#">Duties</a> structure containing duty cycle values.

Here is the caller graph for this function:

**4.20 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
00061 void update_PWM(TIM_HandleTypeDef *htim, Duties duties);
00062
00063 #endif /* PWM_H */

```

## 4.21 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/REFERENCE.h File Reference

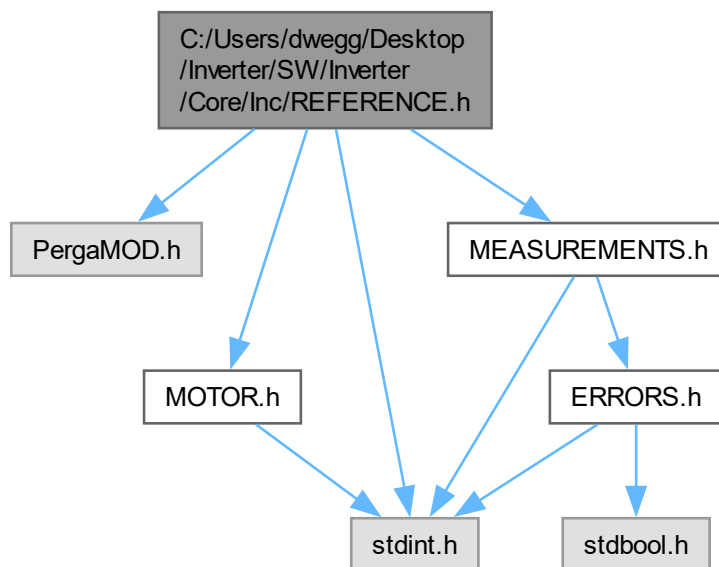
Header file for torque reference handling.

```

#include "PergaMOD.h"
#include "MOTOR.h"
#include "MEASUREMENTS.h"
#include <stdint.h>

```

Include dependency graph for REFERENCE.h:





## 4.21.1 Detailed Description

Header file for torque reference handling.

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>

## 4.21.2 Macro Definition Documentation

### 4.21.2.1 TEMP\_INVERTER\_DERATING

```
#define TEMP_INVERTER_DERATING (OVERTEMPERATURE_INVERTER_TH - 20.0F)
```

Temperature at which linear derating starts for the inverter (20 degC before the fault)

### 4.21.2.2 TEMP\_INVERTER\_MAX

```
#define TEMP_INVERTER_MAX (OVERTEMPERATURE_INVERTER_TH + 10.0F)
```

Temperature at which derating is 0 for the inverter (10 degC more than the fault)

### 4.21.2.3 TEMP\_MOTOR\_DERATING

```
#define TEMP_MOTOR_DERATING (OVERTEMPERATURE_MOTOR_TH - 20.0F)
```

Temperature at which linear derating starts for the motor (20 degC before the fault)

### 4.21.2.4 TEMP\_MOTOR\_MAX

```
#define TEMP_MOTOR_MAX (OVERTEMPERATURE_MOTOR_TH + 10.0F)
```

Temperature at which derating is 0 for the motor (10 degC more than the fault)

## 4.21.3 Function Documentation

### 4.21.3.1 calculate\_derated\_current()

```
float calculate_derated_current (
    float temperature,
    float tempStart,
    float tempMax,
    float iMax )
```

Calculate derated current based on temperature thresholds. It implements a simple linear derating from tempStart to tempMax.

**Parameters**

in	<i>temperature</i>	The current temperature.
in	<i>tempStart</i>	The temperature at which derating starts.
in	<i>tempMax</i>	The temperature at which the current is fully derated to 0.
in	<i>iMax</i>	The maximum current.

**Returns**

The derated current.

Here is the caller graph for this function:

**4.21.3.2 derate\_current\_reference()**

```
float derate_current_reference (
    float tempMotor,
    float tempInverter,
    float iMax )
```

Derate the current reference based on both motor and inverter temperatures.

**Parameters**

in	<i>tempMotor</i>	The motor temperature.
in	<i>tempInverter</i>	The inverter temperature.
in	<i>iMax</i>	The maximum current.

**Returns**

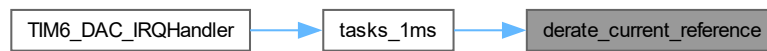
The derated current reference.

Here is the call graph for this function:





Here is the caller graph for this function:



#### 4.21.3.3 handle\_torqueRef()

```

float handle_torqueRef (
    float torqueRefIn,
    int8_t direction,
    float torqueMax,
    float speedMaxRPM,
    float speedMeas,
    volatile pi_struct * loopSpeed )
  
```

Handles torque control based on the reference torque, direction, maximum torque, maximum speed, measured speed, maximum torque rate of change, speed control loop parameters, and sampling time.

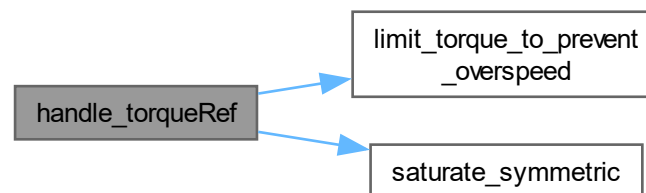
##### Parameters

<i>torqueRefIn</i>	Input reference torque.
<i>direction</i>	Direction of torque (1 for positive torque, -1 for negative torque).
<i>torqueMax</i>	Maximum allowable torque.
<i>speedMaxRPM</i>	Maximum allowable speed in RPM.
<i>speedMeas</i>	Measured speed.
<i>loopSpeed</i>	Speed control loop parameters.

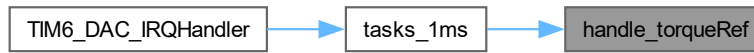
##### Returns

The output torque after handling direction, saturation, and rate limiting.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.21.3.4 limit\_torque\_to\_prevent\_overspeed()

```

float limit_torque_to_prevent_overspeed (
    float speedMaxRPM,
    float speedMeas,
    float torqueRefIn,
    volatile pi_struct * loopSpeed )
  
```

Speed loop acts as a torque saturation, reducing torque in order to limit the maximum speed.

##### Parameters

in	<i>speedMax</i>	The maximum speed value in RPM.
in	<i>speedMeas</i>	The measured speed value in RPM.
in	<i>torqueRefIn</i>	The torque reference value before this saturation.
in	<i>loopSpeed</i>	Pointer to the speed PI controller structure.

##### Returns

torqueRef\_out The limited torque reference value after this saturation.

##### Parameters

in	<i>speedMaxRPM</i>	The maximum speed value in RPM.
in	<i>speedMeas</i>	The measured speed value in RPM.
in	<i>torqueRefIn</i>	The torque reference value before this saturation.
in	<i>loopSpeed</i>	Pointer to the speed PI controller structure.

##### Returns

torqueRefOut The limited torque reference value after this saturation.

Here is the caller graph for this function:



#### 4.21.3.5 saturate\_symmetric()

```
float saturate_symmetric (
    float ref,
    float max )
```

Symmetrically saturate a reference value.

This function symmetrically saturates a reference value based on the maximum allowed value. If the reference value exceeds the maximum allowed value, it is saturated to the maximum value. If the reference value is less than the negative of the maximum allowed value, it is saturated to the negative of the maximum value.

##### Parameters

in	<i>ref</i>	The reference value to saturate.
in	<i>max</i>	The maximum allowed value for saturation.

##### Returns

The saturated reference value.

Here is the caller graph for this function:



#### 4.21.3.6 set\_torque\_direction()

```
float set_torque_direction (
    float torqueRefIn,
    int8_t direction )
```

Set torque direction based on inverter direction.

This function adjusts the torque reference based on the direction of the inverter. If the inverter is set to rotate counterclockwise (CCW), positive torque represents braking. If the inverter is set to rotate clockwise (CW), positive torque represents traction.

##### Parameters

in	<i>torqueRef</i>	The torque reference value to adjust.
in	<i>direction</i>	Pointer to the direction of the inverter (1 for CW, -1 for CCW).

##### Returns

The adjusted torque reference value.

This function adjusts the torque reference based on the desired direction. If the motor is set to rotate counterclockwise (CCW), positive torque represents traction, negative is braking. If the motor is set to rotate clockwise (CW), negative torque represents traction, positive is braking.

#### Parameters

in	<i>torque↔ RefIn</i>	The torque reference value to adjust.
in	<i>direction</i>	Pointer to the direction of the inverter (1 for CW, -1 for CCW).

#### Returns

torqueRefOut The adjusted torque reference value.

### 4.21.4 Variable Documentation

#### 4.21.4.1 torqueRefIn\_left

```
float torqueRefIn_left [extern]
```

#### 4.21.4.2 torqueRefIn\_right

```
float torqueRefIn_right [extern]
```

## 4.22 REFERENCE.h

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

```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020 #ifndef REFERENCE_H
00021 #define REFERENCE_H
00022
00023 #include "PergaMOD.h" // ramp, pi struct
00024 #include "MOTOR.h" // motor struct
00025 #include "MEASUREMENTS.h" // overtemperature defines
00026 #include <stdint.h>
00027
00028 // Define temperature derating thresholds
00029 #define TEMP_MOTOR_DERATING (OVERTEMPERATURE_MOTOR_TH - 20.0F)
00030 #define TEMP_INVERTER_DERATING (OVERTEMPERATURE_INVERTER_TH - 20.0F)
00032 #define TEMP_MOTOR_MAX (OVERTEMPERATURE_MOTOR_TH + 10.0F)
00033 #define TEMP_INVERTER_MAX (OVERTEMPERATURE_INVERTER_TH + 10.0F)
00036 // These variables should be updated via CAN
00037 extern float torqueRefIn_left;
00038 extern float torqueRefIn_right;
00039
00043 typedef struct {
00044     float idRef;
00045     float iqRef;
00046     float isMaxRef;
00047     float torqueRef;
00048 } Reference;
00049
00050
00064 float handle_torqueRef(float torqueRefIn, int8_t direction, float torqueMax, float speedMaxRPM, float
    speedMeas, volatile pi_struct *loopSpeed);
00065
00066
00078 float set_torque_direction(float torqueRef, int8_t direction);
00079
00091 float saturate_symmetric(float ref, float max);
```

```

00092
00101 float limit_torque_to_prevent_overspeed(float speedMax, float speedMeas, float torqueRefIn, volatile
      pi_struct *loopSpeed);
00102
00103
00115 float calculate_derated_current(float temperature, float tempStart, float tempMax, float iMax);
00116
00126 float derate_current_reference(float tempMotor, float tempInverter, float iMax);
00127 #endif /* REFERENCE_H */

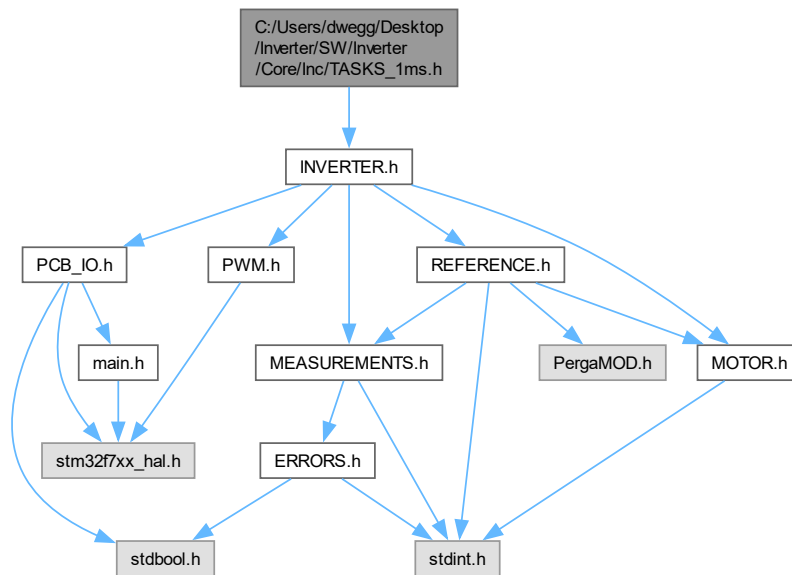
```

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

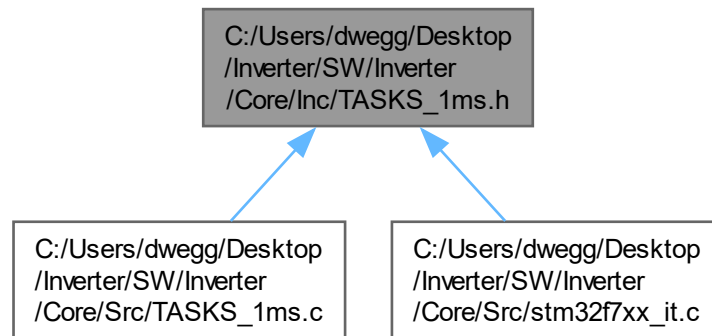
Header file for functions related to tasks executed every 1ms.

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

Include dependency graph for TASKS\_1ms.h:



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



## Functions

- void `tasks_1ms` (void)  
*Function to be executed every 1ms.*
- void `read_temperatures` (void)  
*Function to read temperatures and handle overtemperature faults.*
- void `handle_overtemperature_faults` (volatile `InverterStruct` \*inv)  
*Function to handle overtemperature faults.*

### 4.23.1 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>

### 4.23.2 Function Documentation

#### 4.23.2.1 `handle_overtemperature_faults()`

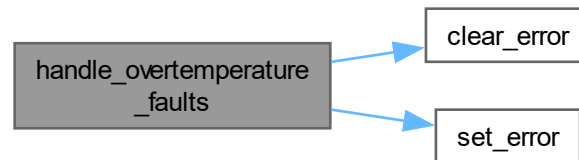
```
void handle_overtemperature_faults (
    volatile InverterStruct * inv )
```

Function to handle overtemperature faults.

## Parameters

<code>in, out</code>	<code>inv</code>	Pointer to the <a href="#">InverterStruct</a> structure.
----------------------	------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:

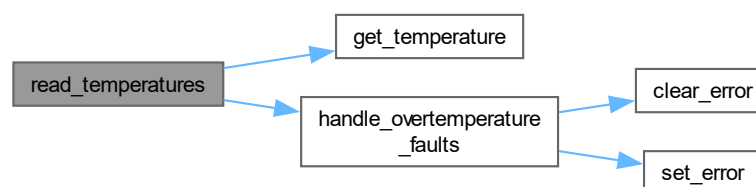


#### 4.23.2.2 read\_temperatures()

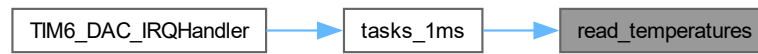
```
void read_temperatures (  
    void )
```

Function to read temperatures and handle overtemperature faults.

Here is the call graph for this function:



Here is the caller graph for this function:



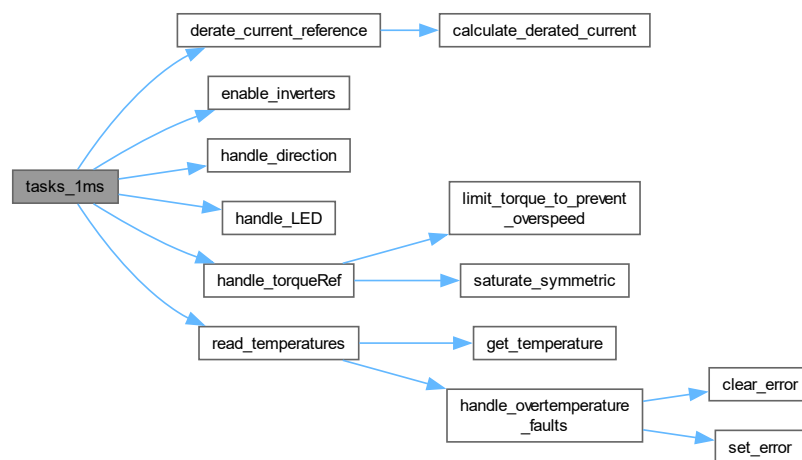
#### 4.23.2.3 tasks\_1ms()

```
void tasks_1ms (
    void )
```

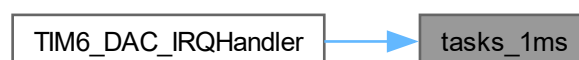
Function to be executed every 1ms.

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 executes all the low priority tasks. Here is the call graph for this function:



Here is the caller graph for this function:





## 4.24 TASKS\_1ms.h

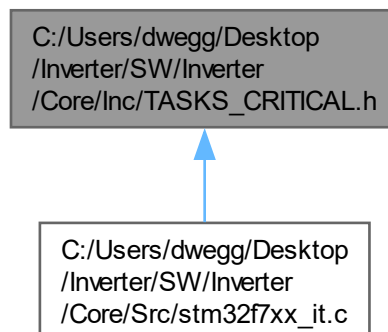
[Go to the documentation of this file.](#)

```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020
00021 #ifndef TASKS_1MS_H
00022 #define TASKS_1MS_H
00023
00024 #include "INVERTER.h" // needs invLeft/invRight
00025
00026
00032 void tasks_1ms(void);
00033
00037 void read_temperatures(void);
00038
00044 void handle_overtemperature_faults(volatile InverterStruct *inv);
00045
00046 #endif /* TASKS_1MS_H */
```

## 4.25 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/TASKS\_↵ CRITICAL.h File Reference

Header file for functions related to tasks executed in each PWM timer interruption.

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



### Functions

- void [tasks\\_critical\\_left](#) ()  
*Function to be executed every TS.*
- void [tasks\\_critical\\_right](#) ()  
*Function to be executed every TS.*

### 4.25.1 Detailed Description

Header file for functions related to tasks executed in each PWM timer interruption.

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>

### 4.25.2 Function Documentation

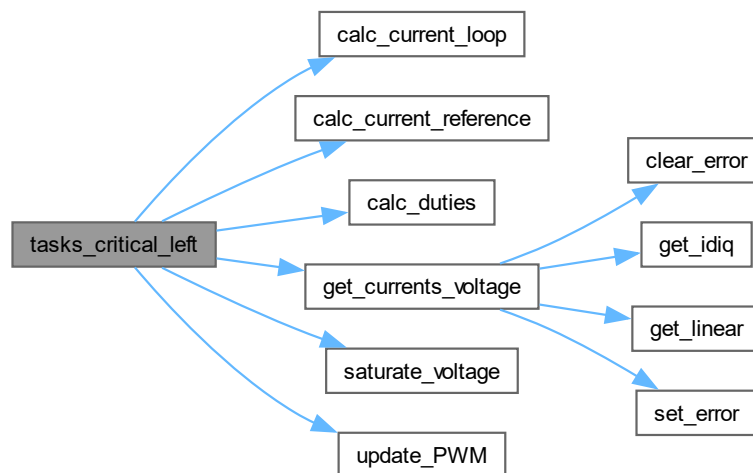
#### 4.25.2.1 tasks\_critical\_left()

```
void tasks_critical_left (
    void )
```

Function to be executed every TS.

This function is called by the TIM1 trigger out handler every TS.

This function is called by the TIM1 trigger handler every TS. Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.25.2.2 tasks\_critical\_right()

```
void tasks_critical_right (
    void )
```

Function to be executed every TS.

This function is called by the TIM8 trigger out handler every TS.

This function is called by the TIM8 trigger handler every TS. Here is the caller graph for this function:



## 4.26 TASKS\_CRITICAL.h

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

```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00025 void tasks_critical_left();
00026
00032 void tasks_critical_right();
```

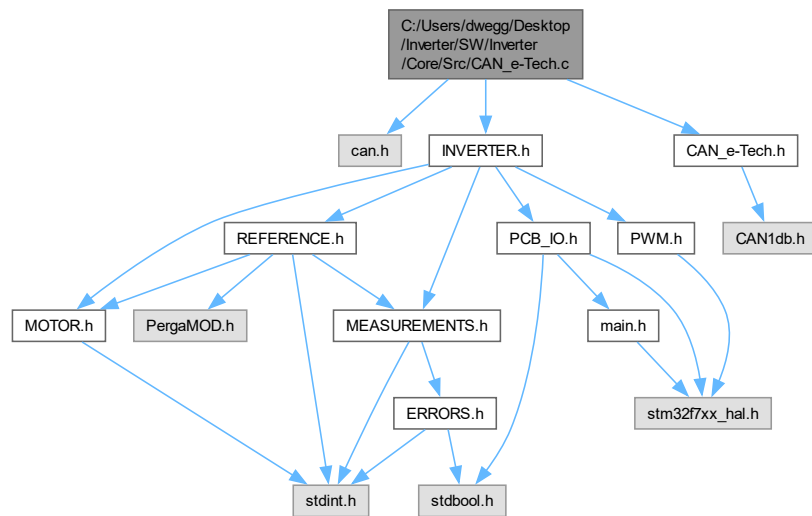
## 4.27 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/CAN\_e- Tech.c File Reference

This file contains functions to handle CAN communication with the car.

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

```
#include "CAN_e-Tech.h"
```

Include dependency graph for CAN\_e-Tech.c:



## Functions

- void `handle_CAN` (CAN\_HandleTypeDef \*hcan)  
*Handle CAN messages.*
- void `send_CAN_message` (CAN\_HandleTypeDef \*hcan, void \*dbc\_msg, const float \*data)  
*Send a CAN message using CAN1db.h information.*

## Variables

- uint8\_t `keepAlive`

### 4.27.1 Detailed Description

This file contains functions to handle CAN communication with the car.

#### 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>

## 4.27.2 Function Documentation

### 4.27.2.1 handle\_CAN()

```
void handle_CAN (
    CAN_HandleTypeDef * hcan )
```

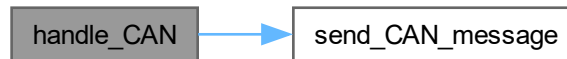
Handle CAN messages.

This function implements the logic to handle received CAN messages.

**Parameters**

<i>hcan</i>	Pointer to the CAN handle structure.
-------------	--------------------------------------

Here is the call graph for this function:



Here is the caller graph for this function:

**4.27.2.2 send\_CAN\_message()**

```

void send_CAN_message (
    CAN_HandleTypeDef * hcan,
    void * dbc_msg,
    const float * data )
  
```

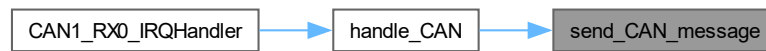
Send a CAN message using CAN1db.h information.

This function prepares and sends a CAN message using information from CAN1db.h.

**Parameters**

<i>hcan</i>	Pointer to the CAN handle structure.
<i>dbc_msg</i>	Pointer to the structure containing CAN message information from CAN1db.h.
<i>data</i>	Pointer to the array of float data to be sent.

Here is the caller graph for this function:



### 4.27.3 Variable Documentation

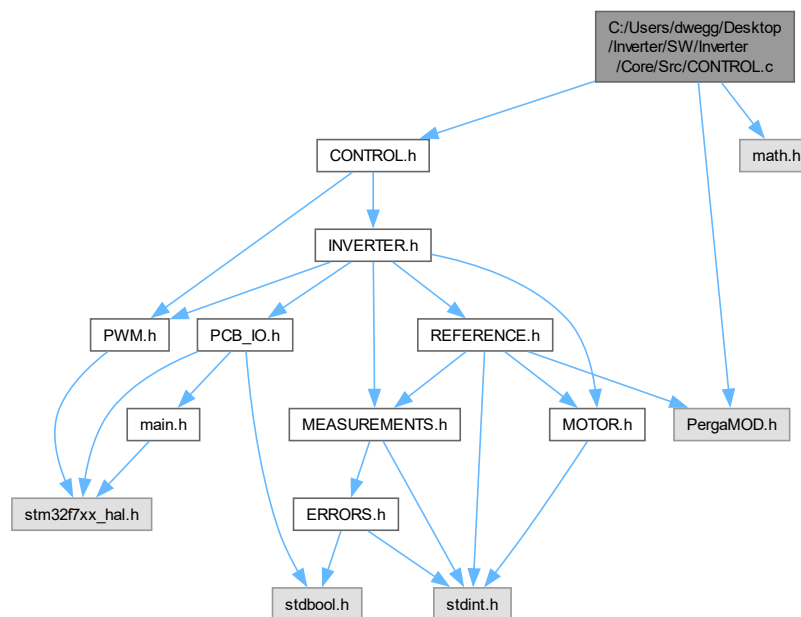
#### 4.27.3.1 keepAlive

```
uint8_t keepAlive
```

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

This file provides code for the control loop.

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



## Functions

- void `calc_current_reference` (`MotorParameters` \*motor, volatile `Reference` \*reference)  
*Calculates the current references using a FOC algorithm. It computes the current vector for the MTPA trajectory and limits the current reference to  $i_{sMaxRef}$  (calculated by derating, starting from the motor's maximum current). The MTPV trajectory is not implemented to save some computation time due to the nature of the motors expected. In order to implement field weakening, an external voltage loop modifying `gammaRef` is needed and should be called inside here. When implementing field weakening, special attention must be put to the torque reference being near 0 or differing from the speed sign (regeneration). A minimum id current must be set for speeds higher than  $V_s/\lambda$ . Study thoroughly, simulate first.*
- void `calc_current_loop` (volatile `InverterStruct` \*inv)  
*Calculates the id-iq loops.*
- void `saturate_voltage` (volatile `InverterStruct` \*inv)  
*Saturates PI output to not surpass DC voltage.*
- void `calc_duties` (float vd, float vq, float vDC, float sinTheta\_e, float cosTheta\_e, volatile `Duties` \*duties)  
*function.*

## 4.28.1 Detailed Description

This file provides code for the control loop.

### Attention

Copyright (c) 2024 David Redondo (@dweggg on 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>

## 4.28.2 Function Documentation

### 4.28.2.1 `calc_current_loop()`

```
void calc_current_loop (
    volatile InverterStruct * inv )
```

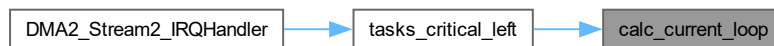
Calculates the id-iq loops.

#### Parameters

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



Here is the caller graph for this function:



#### 4.28.2.2 calc\_current\_reference()

```

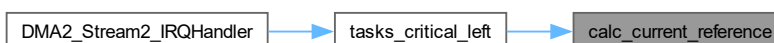
void calc_current_reference (
    MotorParameters * motor,
    volatile Reference * reference )
  
```

Calculates the current references using a FOC algorithm. It computes the current vector for the MTPA trajectory and limits the current reference to isMaxRef (calculated by derating, starting from the motor's maximum current). The MTPV trajectory is not implemented to save some computation time due to the nature of the motors expected. In order to implement field weakening, an external voltage loop modifying gammaRef is needed and should be called inside here. When implementing field weakening, special attention must be put to the torque reference being near 0 or differing from the speed sign (regeneration). A minimum id current must be set for speeds higher than Vs/lambda. Study thoroughly, simulate first.

##### Parameters

in	<i>motor</i>	Pointer to the motor parameters structure.
in, out	<i>reference</i>	Pointer to the reference struct.

Here is the caller graph for this function:



#### 4.28.2.3 calc\_duties()

```

void calc_duties (
    float vd,
    float vq,
    float vDC,
    float sinTheta_e,
    float cosTheta_e,
    volatile Duties * duties )
  
```

function.

This function calculates the inverse Park transform and the duty cycles using SVPWM

**Parameters**

in	<i>vd</i>	Voltage in the d-axis.
in	<i>vq</i>	Voltage in the q-axis.
in	<i>vDC</i>	DC voltage.
in	<i>sinTheta</i> <sub>e</sub>	Electrical angle sine (-1..1)
in	<i>cosTheta</i> <sub>e</sub>	Electrical angle cosine (-1..1)
out	<i>duties</i>	Pointer to the duties structure.

Here is the caller graph for this function:

**4.28.2.4 saturate\_voltage()**

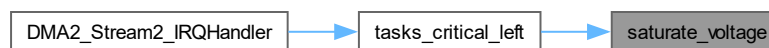
```
void saturate_voltage (
    volatile InverterStruct * inv )
```

Saturates PI output to not surpass DC voltage.

**Parameters**

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

Here is the caller graph for this function:

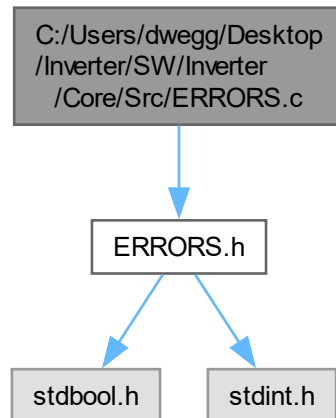


## 4.29 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/ERRORS.c File Reference

Header file for the necessary components to set, read and clear ERRORS.

```
#include "ERRORS.h"
```

Include dependency graph for ERRORS.c:



## Functions

- void `set_error` (volatile void \*data, `InverterError` error)  
*Sets an error in the error field of a data structure.*
- void `clear_error` (volatile void \*data, `InverterError` error)  
*Clears an error in the error field of a data structure.*
- bool `is_error_set` (volatile void \*data, `InverterError` error)  
*Checks if an error is set in the error field of a data structure.*

### 4.29.1 Detailed Description

Header file for the necessary components to set, read and clear ERRORS.

This file contains the necessary components to set, read and clear ERRORS.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg on 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>

## 4.29.2 Function Documentation

### 4.29.2.1 `clear_error()`

```
void clear_error (
    volatile void * data,
    InverterError error )
```

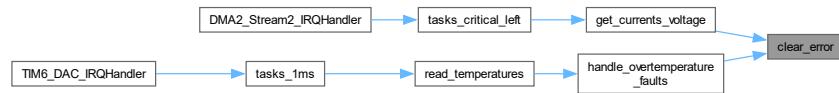
Clears an error in the error field of a data structure.

This function clears the specified error bit in the error field of a data structure.

**Parameters**

out	<i>data</i>	Pointer to the data structure containing the error field.
in	<i>error</i>	The error to be cleared. This should be one of the values from the InverterError enumeration.

Here is the caller graph for this function:

**4.29.2.2 is\_error\_set()**

```

bool is_error_set (
    volatile void * data,
    InverterError error )

```

Checks if an error is set in the error field of a data structure.

This function checks if the specified error bit is set in the error field of a data structure.

**Parameters**

in	<i>data</i>	Pointer to the data structure containing the error field.
in	<i>error</i>	The error to be checked. This should be one of the values from the InverterError enumeration.

**Returns**

true if the specified error is set, false otherwise.

**4.29.2.3 set\_error()**

```

void set_error (
    volatile void * data,
    InverterError error )

```

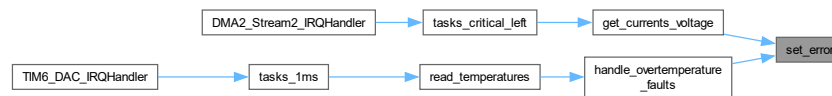
Sets an error in the error field of a data structure.

This function sets the specified error bit in the error field of a data structure.

**Parameters**

out	<i>data</i>	Pointer to the data structure containing the error field.
in	<i>error</i>	The error to be set. This should be one of the values from the InverterError enumeration.

Here is the caller graph for this function:

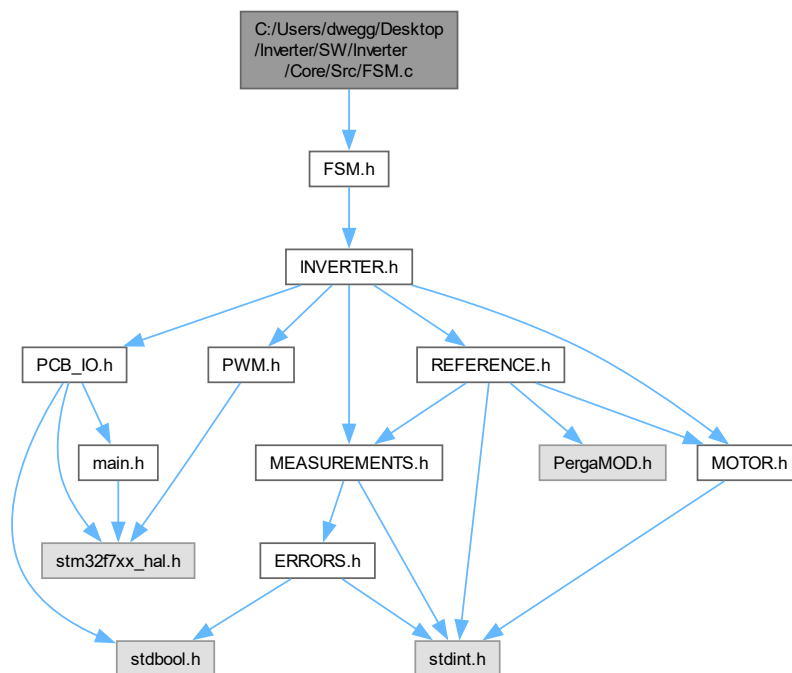


## 4.30 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/FSM.c File Reference

This file provides code for the inverter Finite State Machine.

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

Include dependency graph for FSM.c:



### Functions

- void `eval_inv_FSM` (volatile `InverterStruct` \*inv)  
Execute the finite state machine for inverter.

### 4.30.1 Detailed Description

This file provides code for the inverter Finite State Machine.

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>

### 4.30.2 Function Documentation

#### 4.30.2.1 eval\_inv\_FSM()

```
void eval_inv_FSM (  
    volatile 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

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

Here is the caller graph for this function:



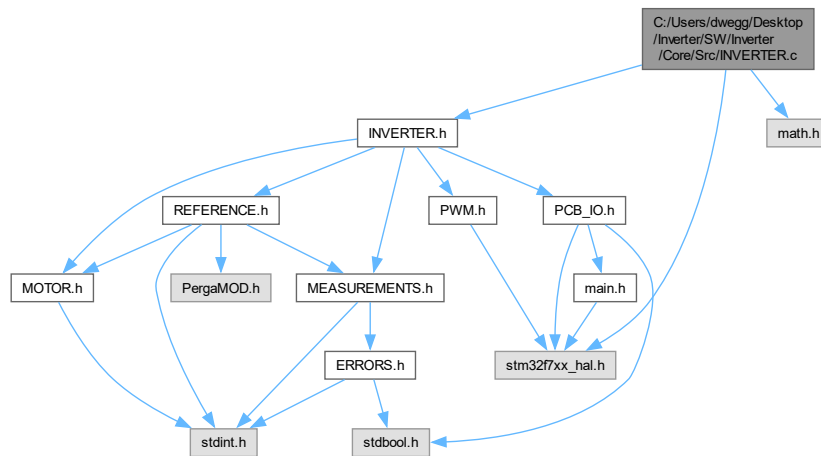
## 4.31 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/INVERTER.c File Reference

This file provides code for the inverter struct.

```
#include "INVERTER.h"  
#include "stm32f7xx_hal.h"
```

```
#include <math.h>
```

Include dependency graph for INVERTER.c:



## Functions

- void `initialize_inverter` (volatile `InverterStruct` \*inv, `LED` \*led, `GPIO_TypeDef` \*enable\_port, uint16\_t enable\_pin, `TIM_HandleTypeDef` \*htim, `ADC_HandleTypeDef` \*hadc, `MotorParameters` \*motor, volatile uint16\_t \*rawADC)  
*Initialize the inverter.*
- void `init_control_loops` (volatile `InverterStruct` \*inv, `MotorParameters` \*motor)  
*Initializes the PI controllers.*
- void `enable_control_loops` (volatile `InverterStruct` \*inv)  
*Enables the PI controllers.*
- void `disable_control_loops` (volatile `InverterStruct` \*inv)  
*Disables the PI controllers.*

## Variables

- volatile `InverterStruct` `inverter_left` = {0}  
*Left inverter structure.*
- volatile `InverterStruct` `inverter_right` = {0}  
*Right inverter structure.*

### 4.31.1 Detailed Description

This file provides code for the inverter struct.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg on 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>



## 4.31.2 Function Documentation

### 4.31.2.1 disable\_control\_loops()

```
void disable_control_loops (
    volatile InverterStruct * inv )
```

Disables the PI controllers.

#### Parameters

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

### 4.31.2.2 enable\_control\_loops()

```
void enable_control_loops (
    volatile InverterStruct * inv )
```

Enables the PI controllers.

#### Parameters

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

### 4.31.2.3 init\_control\_loops()

```
void init_control_loops (
    volatile InverterStruct * inv,
    MotorParameters * motor )
```

Initializes the PI controllers.

Initializes the id-iq current control PI controllers.

#### Parameters

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

Here is the caller graph for this function:



#### 4.31.2.4 initialize\_inverter()

```
void initialize_inverter (
    volatile InverterStruct * inv,
    LED * led,
    GPIO_TypeDef * enable_port,
    uint16_t enable_pin,
    TIM_HandleTypeDef * htim,
    ADC_HandleTypeDef * hadc,
    MotorParameters * motor,
    volatile uint16_t * rawADC )
```

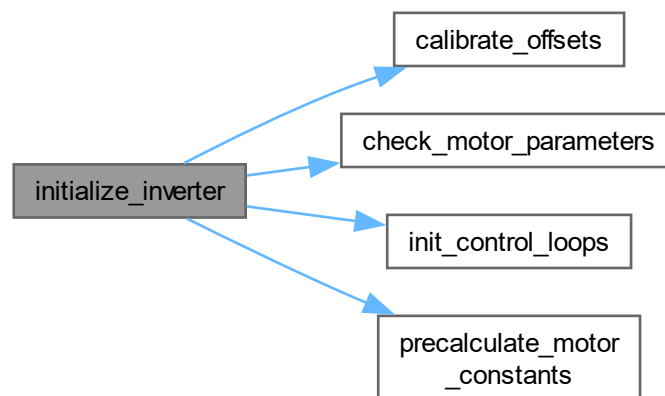
Initialize the inverter.

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

##### Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



### 4.31.3 Variable Documentation

#### 4.31.3.1 inverter\_left

```
volatile InverterStruct inverter_left = {0}
```

Left inverter structure.

External declaration of the left inverter structure.

#### 4.31.3.2 inverter\_right

```
volatile InverterStruct inverter_right = {0}
```

Right inverter structure.

External declaration of the right inverter structure.

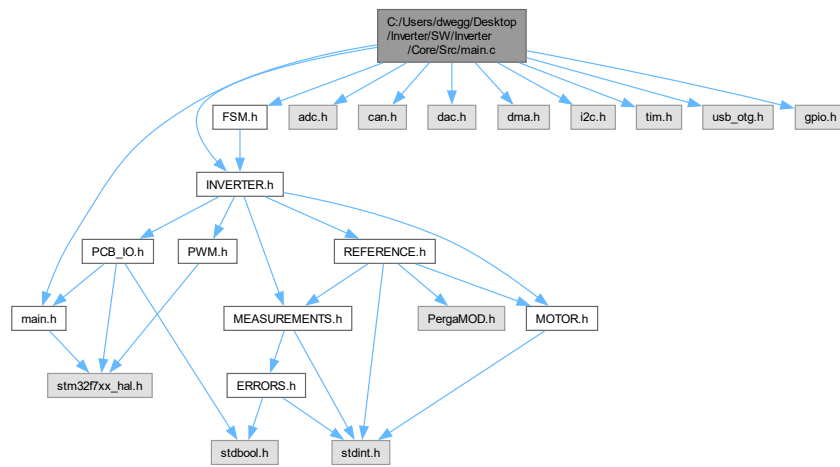
## 4.32 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/main.c File Reference

: 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 "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.*

### 4.32.1 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.

### 4.32.2 Function Documentation

#### 4.32.2.1 Error\_Handler()

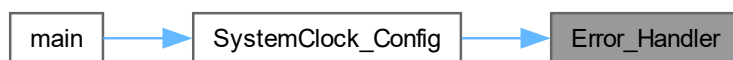
```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

## Return values

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

Here is the caller graph for this function:



## 4.32.2.2 main()

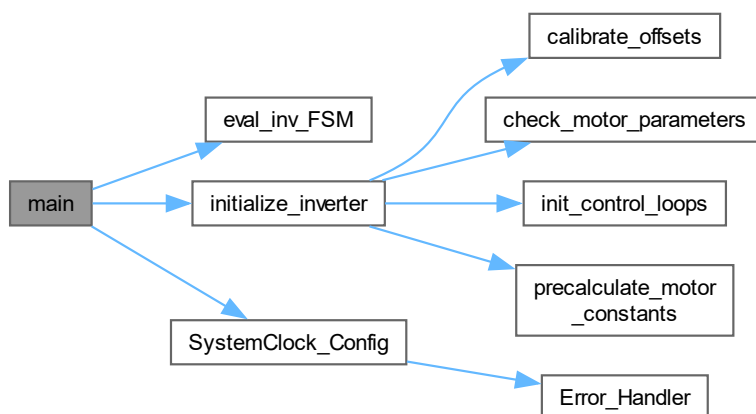
```
int main (
    void )
```

The application entry point.

## Return values

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

Here is the call graph for this function:



## 4.32.2.3 SystemClock\_Config()

```
void SystemClock_Config (
```

```
void )
```

System Clock Configuration.

#### Return values

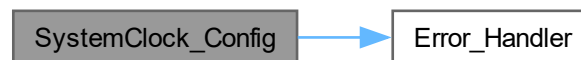
None	
------	--

Configure the main internal regulator output voltage

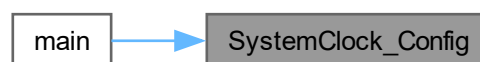
Initializes the RCC Oscillators according to the specified parameters in the RCC\_OscInitTypeDef structure.

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:



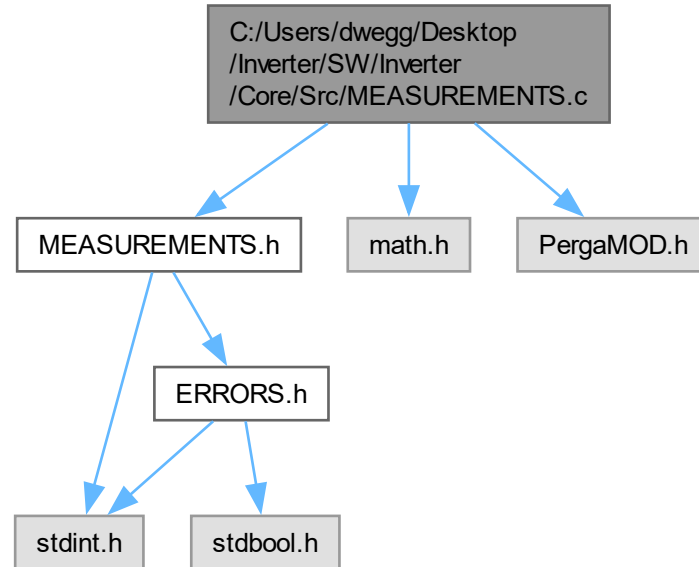
### 4.33 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/↵ MEASUREMENTS.c File Reference

This file provides functions for handling measurements.

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

```
#include <PergaMOD.h>
```

Include dependency graph for MEASUREMENTS.c:



## Functions

- uint8\_t [get\\_currents\\_voltage](#) (volatile uint16\_t ADC\_raw[], volatile [Analog](#) \*analog, volatile [Feedback](#) \*feedback, volatile [InverterError](#) \*errors, float sinTheta\_e, float cosTheta\_e)  
*Get electrical ADC measurements.*
- float [get\\_linear](#) (uint32\_t bits, float slope, float offset)  
*Convert ADC reading to physical measurement with linear response.*
- void [get\\_idiq](#) (float ia, float ib, float ic, float sinTheta\_e, float cosTheta\_e, float \*idMeas, float \*iqMeas)  
*Computes d-q currents from current measurements and electrical angle.*
- float [get\\_temperature](#) (uint32\_t bits, const float tempLUT[])  
*Retrieves temperature from a lookup table based on ADC bits.*
- void [calibrate\\_offsets](#) (volatile uint16\_t rawADC[], volatile float currentOffsets[], uint32\_t numSamples)  
*Calibrate the current sensor offsets.*

## Variables

- const float [templInverterLUT](#) [] = {-2.45, -2.44, -2.44, -2.43, -2.42, -2.42, -2.41, -2.41, -2.40, -2.39, -2.39, -2.38, -2.37, -2.37, -2.36, -2.36, -2.35, -2.34, -2.34, -2.33, -2.32, -2.32, -2.31, -2.31, -2.30, -2.29, -2.29, -2.28, -2.27, -2.27, -2.26, -2.26, -2.25, -2.24, -2.24, -2.23, -2.22, -2.22, -2.21, -2.20, -2.20, -2.19, -2.19, -2.18, -2.17, -2.17, -2.16, -2.15, -2.15, -2.14, -2.14, -2.13, -2.12, -2.12, -2.11, -2.10, -2.10, -2.09, -2.08, -2.08, -2.07, -2.07, -2.06, -2.05, -2.05, -2.04, -2.03, -2.03, -2.02, -2.01, -2.01, -2.00, -2.00, -1.99, -1.98, -1.98, -1.97, -1.96, -1.96, -1.95, -1.94, -1.94, -1.93, -1.93, -1.92, -1.91, -1.91, -1.90, -1.89, -1.89, -1.88, -1.87, -1.87, -1.86, -1.86, -1.85, -1.84, -1.84, -1.83, -1.82, -1.82, -1.81, -1.80, -1.80, -1.79, -1.78, -1.78, -1.77, -1.77, -1.76, -1.75, -1.75, -1.74, -1.73, -1.73, -1.72, -1.71, -1.71, -1.70, -1.69, -1.69, -1.68, -1.67, -1.67, -1.66, -1.66, -1.65, -1.64, -1.64, -1.63, -1.62, -1.62, -1.61, -1.60, -1.60, -1.59, -1.58, -1.58, -1.57, -1.56, -1.56, -1.55, -1.54, -1.54, -1.53, -1.53, -1.52,

-1.51, -1.51, -1.50, -1.49, -1.49, -1.48, -1.47, -1.47, -1.46, -1.45, -1.45, -1.44, -1.43, -1.43, -1.42, -1.41, -1.41, -1.40, -1.39, -1.39, -1.38, -1.37, -1.37, -1.36, -1.36, -1.35, -1.34, -1.34, -1.33, -1.32, -1.32, -1.31, -1.30, -1.30, -1.29, -1.28, -1.28, -1.27, -1.26, -1.26, -1.25, -1.24, -1.24, -1.23, -1.22, -1.22, -1.21, -1.20, -1.20, -1.19, -1.18, -1.18, -1.17, -1.16, -1.16, -1.15, -1.14, -1.14, -1.13, -1.12, -1.12, -1.11, -1.10, -1.10, -1.09, -1.08, -1.08, -1.07, -1.06, -1.06, -1.05, -1.04, -1.04, -1.03, -1.02, -1.02, -1.01, -1.00, -1.00, -0.99, -0.98, -0.98, -0.97, -0.96, -0.96, -0.95, -0.94, -0.94, -0.93, -0.92, -0.92, -0.91, -0.90, -0.90, -0.89, -0.88, -0.88, -0.87, -0.86, -0.86, -0.85, -0.84, -0.84, -0.83, -0.82, -0.82, -0.81, -0.80, -0.80, -0.79, -0.78, -0.78, -0.77, -0.76, -0.76, -0.75, -0.74, -0.73, -0.73, -0.72, -0.71, -0.71, -0.70, -0.69, -0.69, -0.68, -0.67, -0.67, -0.66, -0.65, -0.65, -0.64, -0.63, -0.63, -0.62, -0.61, -0.61, -0.60, -0.59, -0.59, -0.58, -0.57, -0.56, -0.56, -0.55, -0.54, -0.54, -0.53, -0.52, -0.52, -0.51, -0.50, -0.50, -0.49, -0.48, -0.48, -0.47, -0.46, -0.46, -0.45, -0.44, -0.43, -0.43, -0.42, -0.41, -0.41, -0.40, -0.39, -0.39, -0.38, -0.37, -0.37, -0.36, -0.35, -0.35, -0.34, -0.33, -0.32, -0.32, -0.31, -0.30, -0.30, -0.29, -0.28, -0.28, -0.27, -0.26, -0.26, -0.25, -0.24, -0.23, -0.23, -0.22, -0.21, -0.21, -0.20, -0.19, -0.19, -0.18, -0.17, -0.17, -0.16, -0.15, -0.14, -0.14, -0.13, -0.12, -0.12, -0.11, -0.10, -0.10, -0.09, -0.08, -0.07, -0.07, -0.06, -0.05, -0.05, -0.04, -0.03, -0.03, -0.02, -0.01, -0.00, 0.00, 0.01, 0.02, 0.02, 0.03, 0.04, 0.04, 0.05, 0.06, 0.07, 0.07, 0.08, 0.09, 0.09, 0.10, 0.11, 0.12, 0.12, 0.13, 0.14, 0.14, 0.15, 0.16, 0.16, 0.17, 0.18, 0.19, 0.19, 0.20, 0.21, 0.21, 0.22, 0.23, 0.24, 0.24, 0.25, 0.26, 0.26, 0.27, 0.28, 0.29, 0.29, 0.30, 0.31, 0.31, 0.32, 0.33, 0.34, 0.34, 0.35, 0.36, 0.36, 0.37, 0.38, 0.39, 0.39, 0.40, 0.41, 0.41, 0.42, 0.43, 0.44, 0.44, 0.45, 0.46, 0.46, 0.47, 0.48, 0.49, 0.49, 0.50, 0.51, 0.51, 0.52, 0.53, 0.54, 0.54, 0.55, 0.56, 0.56, 0.57, 0.58, 0.59, 0.59, 0.60, 0.61, 0.61, 0.62, 0.63, 0.64, 0.64, 0.65, 0.66, 0.67, 0.67, 0.68, 0.69, 0.69, 0.70, 0.71, 0.72, 0.72, 0.73, 0.74, 0.75, 0.75, 0.76, 0.77, 0.77, 0.78, 0.79, 0.80, 0.80, 0.81, 0.82, 0.83, 0.83, 0.84, 0.85, 0.85, 0.86, 0.87, 0.88, 0.88, 0.89, 0.90, 0.91, 0.91, 0.92, 0.93, 0.94, 0.94, 0.95, 0.96, 0.96, 0.97, 0.98, 0.99, 0.99, 1.00, 1.01, 1.02, 1.02, 1.03, 1.04, 1.05, 1.05, 1.06, 1.07, 1.08, 1.08, 1.09, 1.10, 1.10, 1.11, 1.12, 1.13, 1.13, 1.14, 1.15, 1.16, 1.16, 1.17, 1.18, 1.19, 1.19, 1.20, 1.21, 1.22, 1.22, 1.23, 1.24, 1.25, 1.25, 1.26, 1.27, 1.28, 1.28, 1.29, 1.30, 1.31, 1.31, 1.32, 1.33, 1.34, 1.34, 1.35, 1.36, 1.37, 1.37, 1.38, 1.39, 1.40, 1.40, 1.41, 1.42, 1.43, 1.43, 1.44, 1.45, 1.46, 1.46, 1.47, 1.48, 1.49, 1.49, 1.50, 1.51, 1.52, 1.52, 1.53, 1.54, 1.55, 1.55, 1.56, 1.57, 1.58, 1.58, 1.59, 1.60, 1.61, 1.61, 1.62, 1.63, 1.64, 1.64, 1.65, 1.66, 1.67, 1.67, 1.68, 1.69, 1.70, 1.71, 1.71, 1.72, 1.73, 1.74, 1.74, 1.75, 1.76, 1.77, 1.77, 1.78, 1.79, 1.80, 1.80, 1.81, 1.82, 1.83, 1.84, 1.84, 1.85, 1.86, 1.87, 1.87, 1.88, 1.89, 1.90, 1.90, 1.91, 1.92, 1.93, 1.93, 1.94, 1.95, 1.96, 1.97, 1.97, 1.98, 1.99, 2.00, 2.00, 2.01, 2.02, 2.03, 2.04, 2.04, 2.05, 2.06, 2.07, 2.07, 2.08, 2.09, 2.10, 2.10, 2.11, 2.12, 2.13, 2.14, 2.14, 2.15, 2.16, 2.17, 2.17, 2.18, 2.19, 2.20, 2.21, 2.21, 2.22, 2.23, 2.24, 2.25, 2.25, 2.26, 2.27, 2.28, 2.28, 2.29, 2.30, 2.31, 2.32, 2.32, 2.33, 2.34, 2.35, 2.35, 2.36, 2.37, 2.38, 2.39, 2.39, 2.40, 2.41, 2.42, 2.43, 2.43, 2.44, 2.45, 2.46, 2.46, 2.47, 2.48, 2.49, 2.50, 2.50, 2.51, 2.52, 2.53, 2.54, 2.54, 2.55, 2.56, 2.57, 2.58, 2.58, 2.59, 2.60, 2.61, 2.62, 2.62, 2.63, 2.64, 2.65, 2.66, 2.66, 2.67, 2.68, 2.69, 2.70, 2.70, 2.71, 2.72, 2.73, 2.74, 2.74, 2.75, 2.76, 2.77, 2.78, 2.78, 2.79, 2.80, 2.81, 2.82, 2.82, 2.83, 2.84, 2.85, 2.86, 2.86, 2.87, 2.88, 2.89, 2.90, 2.90, 2.91, 2.92, 2.93, 2.94, 2.94, 2.95, 2.96, 2.97, 2.98, 2.98, 2.99, 3.00, 3.01, 3.02, 3.02, 3.03, 3.04, 3.05, 3.06, 3.07, 3.07, 3.08, 3.09, 3.10, 3.11, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.16, 3.17, 3.18, 3.19, 3.20, 3.20, 3.21, 3.22, 3.23, 3.24, 3.24, 3.25, 3.26, 3.27, 3.28, 3.29, 3.29, 3.30, 3.31, 3.32, 3.33, 3.34, 3.34, 3.35, 3.36, 3.37, 3.38, 3.38, 3.39, 3.40, 3.41, 3.42, 3.43, 3.43, 3.44, 3.45, 3.46, 3.47, 3.48, 3.48, 3.49, 3.50, 3.51, 3.52, 3.53, 3.53, 3.54, 3.55, 3.56, 3.57, 3.58, 3.58, 3.59, 3.60, 3.61, 3.62, 3.63, 3.63, 3.64, 3.65, 3.66, 3.67, 3.68, 3.68, 3.69, 3.70, 3.71, 3.72, 3.73, 3.73, 3.74, 3.75, 3.76, 3.77, 3.78, 3.78, 3.79, 3.80, 3.81, 3.82, 3.83, 3.83, 3.84, 3.85, 3.86, 3.87, 3.88, 3.89, 3.89, 3.90, 3.91, 3.92, 3.93, 3.94, 3.94, 3.95, 3.96, 3.97, 3.98, 3.99, 4.00, 4.00, 4.01, 4.02, 4.03, 4.04, 4.05, 4.05, 4.06, 4.07, 4.08, 4.09, 4.10, 4.11, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.17, 4.18, 4.19, 4.20, 4.21, 4.22, 4.23, 4.23, 4.24, 4.25, 4.26, 4.27, 4.28, 4.29, 4.29, 4.30, 4.31, 4.32, 4.33, 4.34, 4.35, 4.35, 4.36, 4.37, 4.38, 4.39, 4.40, 4.41, 4.42, 4.42, 4.43, 4.44, 4.45, 4.46, 4.47, 4.48, 4.48, 4.49, 4.50, 4.51, 4.52, 4.53, 4.54, 4.55, 4.55, 4.56, 4.57, 4.58, 4.59, 4.60, 4.61, 4.62, 4.62, 4.63, 4.64, 4.65, 4.66, 4.67, 4.68, 4.69, 4.69, 4.70, 4.71, 4.72, 4.73, 4.74, 4.75, 4.76, 4.76, 4.77, 4.78, 4.79, 4.80, 4.81, 4.82, 4.83, 4.83, 4.84, 4.85, 4.86, 4.87, 4.88, 4.89, 4.90, 4.91, 4.91, 4.92, 4.93, 4.94, 4.95, 4.96, 4.97, 4.98, 4.99, 4.99, 5.00, 5.01, 5.02, 5.03, 5.04, 5.05, 5.06, 5.07, 5.07, 5.08, 5.09, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.16, 5.17, 5.18, 5.19, 5.20, 5.21, 5.22, 5.23, 5.24, 5.24, 5.25, 5.26, 5.27, 5.28, 5.29, 5.30, 5.31, 5.32, 5.33, 5.34, 5.34, 5.35, 5.36, 5.37, 5.38, 5.39, 5.40, 5.41, 5.42, 5.43, 5.43, 5.44, 5.45, 5.46, 5.47, 5.48, 5.49, 5.50, 5.51, 5.52, 5.53, 5.53, 5.54, 5.55, 5.56, 5.57, 5.58, 5.59, 5.60, 5.61, 5.62, 5.63, 5.64, 5.64, 5.65, 5.66, 5.67, 5.68, 5.69, 5.70, 5.71, 5.72, 5.73, 5.74, 5.75, 5.75, 5.76, 5.77, 5.78, 5.79, 5.80, 5.81, 5.82, 5.83, 5.84, 5.85, 5.86, 5.87, 5.88, 5.88, 5.89, 5.90, 5.91, 5.92, 5.93, 5.94, 5.95, 5.96, 5.97, 5.98, 5.99, 6.00, 6.01, 6.01, 6.02, 6.03, 6.04, 6.05, 6.06, 6.07, 6.08, 6.09, 6.10, 6.11, 6.12, 6.13, 6.14, 6.15, 6.16, 6.16, 6.17, 6.18, 6.19, 6.20, 6.21, 6.22, 6.23, 6.24, 6.25, 6.26, 6.27, 6.28, 6.29, 6.30, 6.31, 6.32, 6.32, 6.33, 6.34, 6.35, 6.36, 6.37, 6.38, 6.39, 6.40, 6.41, 6.42, 6.43, 6.44, 6.45, 6.46, 6.47, 6.48, 6.49, 6.50, 6.51, 6.51, 6.52, 6.53, 6.54, 6.55, 6.56, 6.57, 6.58, 6.59, 6.60, 6.61, 6.62, 6.63, 6.64, 6.65, 6.66, 6.67, 6.68, 6.69, 6.70, 6.71, 6.72, 6.73, 6.74, 6.75, 6.75, 6.76, 6.77, 6.78, 6.79, 6.80, 6.81, 6.82, 6.83, 6.84, 6.85, 6.86, 6.87, 6.88, 6.89, 6.90, 6.91, 6.92, 6.93, 6.94, 6.95, 6.96, 6.97, 6.98, 6.99, 7.00, 7.01, 7.02,



7.03, 7.04, 7.05, 7.06, 7.07, 7.08, 7.09, 7.10, 7.11, 7.12, 7.12, 7.13, 7.14, 7.15, 7.16, 7.17, 7.18, 7.19, 7.20, 7.21, 7.22, 7.23, 7.24, 7.25, 7.26, 7.27, 7.28, 7.29, 7.30, 7.31, 7.32, 7.33, 7.34, 7.35, 7.36, 7.37, 7.38, 7.39, 7.40, 7.41, 7.42, 7.43, 7.44, 7.45, 7.46, 7.47, 7.48, 7.49, 7.50, 7.51, 7.52, 7.53, 7.54, 7.55, 7.56, 7.57, 7.58, 7.59, 7.60, 7.61, 7.62, 7.63, 7.64, 7.65, 7.66, 7.67, 7.68, 7.69, 7.70, 7.71, 7.72, 7.73, 7.74, 7.75, 7.76, 7.77, 7.78, 7.79, 7.80, 7.81, 7.82, 7.83, 7.84, 7.85, 7.86, 7.87, 7.88, 7.89, 7.91, 7.92, 7.93, 7.94, 7.95, 7.96, 7.97, 7.98, 7.99, 8.00, 8.01, 8.02, 8.03, 8.04, 8.05, 8.06, 8.07, 8.08, 8.09, 8.10, 8.11, 8.12, 8.13, 8.14, 8.15, 8.16, 8.17, 8.18, 8.19, 8.20, 8.21, 8.22, 8.23, 8.24, 8.25, 8.26, 8.27, 8.29, 8.30, 8.31, 8.32, 8.33, 8.34, 8.35, 8.36, 8.37, 8.38, 8.39, 8.40, 8.41, 8.42, 8.43, 8.44, 8.45, 8.46, 8.47, 8.48, 8.49, 8.50, 8.51, 8.52, 8.54, 8.55, 8.56, 8.57, 8.58, 8.59, 8.60, 8.61, 8.62, 8.63, 8.64, 8.65, 8.66, 8.67, 8.68, 8.69, 8.70, 8.71, 8.72, 8.74, 8.75, 8.76, 8.77, 8.78, 8.79, 8.80, 8.81, 8.82, 8.83, 8.84, 8.85, 8.86, 8.87, 8.88, 8.89, 8.91, 8.92, 8.93, 8.94, 8.95, 8.96, 8.97, 8.98, 8.99, 9.00, 9.01, 9.02, 9.03, 9.04, 9.06, 9.07, 9.08, 9.09, 9.10, 9.11, 9.12, 9.13, 9.14, 9.15, 9.16, 9.17, 9.18, 9.20, 9.21, 9.22, 9.23, 9.24, 9.25, 9.26, 9.27, 9.28, 9.29, 9.30, 9.31, 9.33, 9.34, 9.35, 9.36, 9.37, 9.38, 9.39, 9.40, 9.41, 9.42, 9.43, 9.45, 9.46, 9.47, 9.48, 9.49, 9.50, 9.51, 9.52, 9.53, 9.54, 9.55, 9.57, 9.58, 9.59, 9.60, 9.61, 9.62, 9.63, 9.64, 9.65, 9.66, 9.68, 9.69, 9.70, 9.71, 9.72, 9.73, 9.74, 9.75, 9.76, 9.78, 9.79, 9.80, 9.81, 9.82, 9.83, 9.84, 9.85, 9.86, 9.88, 9.89, 9.90, 9.91, 9.92, 9.93, 9.94, 9.95, 9.96, 9.98, 9.99, 10.00, 10.01, 10.02, 10.03, 10.04, 10.05, 10.07, 10.08, 10.09, 10.10, 10.11, 10.12, 10.13, 10.14, 10.16, 10.17, 10.18, 10.19, 10.20, 10.21, 10.22, 10.24, 10.25, 10.26, 10.27, 10.28, 10.29, 10.30, 10.31, 10.33, 10.34, 10.35, 10.36, 10.37, 10.38, 10.39, 10.41, 10.42, 10.43, 10.44, 10.45, 10.46, 10.47, 10.49, 10.50, 10.51, 10.52, 10.53, 10.54, 10.55, 10.57, 10.58, 10.59, 10.60, 10.61, 10.62, 10.64, 10.65, 10.66, 10.67, 10.68, 10.69, 10.70, 10.72, 10.73, 10.74, 10.75, 10.76, 10.77, 10.79, 10.80, 10.81, 10.82, 10.83, 10.84, 10.86, 10.87, 10.88, 10.89, 10.90, 10.91, 10.93, 10.94, 10.95, 10.96, 10.97, 10.98, 11.00, 11.01, 11.02, 11.03, 11.04, 11.05, 11.07, 11.08, 11.09, 11.10, 11.11, 11.13, 11.14, 11.15, 11.16, 11.17, 11.18, 11.20, 11.21, 11.22, 11.23, 11.24, 11.26, 11.27, 11.28, 11.29, 11.30, 11.32, 11.33, 11.34, 11.35, 11.36, 11.37, 11.39, 11.40, 11.41, 11.42, 11.43, 11.45, 11.46, 11.47, 11.48, 11.49, 11.51, 11.52, 11.53, 11.54, 11.55, 11.57, 11.58, 11.59, 11.60, 11.61, 11.63, 11.64, 11.65, 11.66, 11.68, 11.69, 11.70, 11.71, 11.72, 11.74, 11.75, 11.76, 11.77, 11.78, 11.80, 11.81, 11.82, 11.83, 11.85, 11.86, 11.87, 11.88, 11.89, 11.91, 11.92, 11.93, 11.94, 11.96, 11.97, 11.98, 11.99, 12.00, 12.02, 12.03, 12.04, 12.05, 12.07, 12.08, 12.09, 12.10, 12.11, 12.13, 12.14, 12.15, 12.16, 12.18, 12.19, 12.20, 12.21, 12.23, 12.24, 12.25, 12.26, 12.28, 12.29, 12.30, 12.31, 12.33, 12.34, 12.35, 12.36, 12.38, 12.39, 12.40, 12.41, 12.43, 12.44, 12.45, 12.46, 12.48, 12.49, 12.50, 12.51, 12.53, 12.54, 12.55, 12.56, 12.58, 12.59, 12.60, 12.61, 12.63, 12.64, 12.65, 12.66, 12.68, 12.69, 12.70, 12.72, 12.73, 12.74, 12.75, 12.77, 12.78, 12.79, 12.80, 12.82, 12.83, 12.84, 12.86, 12.87, 12.88, 12.89, 12.91, 12.92, 12.93, 12.94, 12.96, 12.97, 12.98, 13.00, 13.01, 13.02, 13.03, 13.05, 13.06, 13.07, 13.09, 13.10, 13.11, 13.12, 13.14, 13.15, 13.16, 13.18, 13.19, 13.20, 13.22, 13.23, 13.24, 13.25, 13.27, 13.28, 13.29, 13.31, 13.32, 13.33, 13.35, 13.36, 13.37, 13.38, 13.40, 13.41, 13.42, 13.44, 13.45, 13.46, 13.48, 13.49, 13.50, 13.52, 13.53, 13.54, 13.55, 13.57, 13.58, 13.59, 13.61, 13.62, 13.63, 13.65, 13.66, 13.67, 13.69, 13.70, 13.71, 13.73, 13.74, 13.75, 13.77, 13.78, 13.79, 13.81, 13.82, 13.83, 13.85, 13.86, 13.87, 13.89, 13.90, 13.91, 13.93, 13.94, 13.95, 13.97, 13.98, 13.99, 14.01, 14.02, 14.03, 14.05, 14.06, 14.07, 14.09, 14.10, 14.11, 14.13, 14.14, 14.16, 14.17, 14.18, 14.20, 14.21, 14.22, 14.24, 14.25, 14.26, 14.28, 14.29, 14.30, 14.32, 14.33, 14.35, 14.36, 14.37, 14.39, 14.40, 14.41, 14.43, 14.44, 14.45, 14.47, 14.48, 14.50, 14.51, 14.52, 14.54, 14.55, 14.56, 14.58, 14.59, 14.61, 14.62, 14.63, 14.65, 14.66, 14.67, 14.69, 14.70, 14.72, 14.73, 14.74, 14.76, 14.77, 14.79, 14.80, 14.81, 14.83, 14.84, 14.86, 14.87, 14.88, 14.90, 14.91, 14.93, 14.94, 14.95, 14.97, 14.98, 15.00, 15.01, 15.02, 15.04, 15.05, 15.07, 15.08, 15.09, 15.11, 15.12, 15.14, 15.15, 15.16, 15.18, 15.19, 15.21, 15.22, 15.24, 15.25, 15.26, 15.28, 15.29, 15.31, 15.32, 15.33, 15.35, 15.36, 15.38, 15.39, 15.41, 15.42, 15.43, 15.45, 15.46, 15.48, 15.49, 15.51, 15.52, 15.54, 15.55, 15.56, 15.58, 15.59, 15.61, 15.62, 15.64, 15.65, 15.66, 15.68, 15.69, 15.71, 15.72, 15.74, 15.75, 15.77, 15.78, 15.80, 15.81, 15.82, 15.84, 15.85, 15.87, 15.88, 15.90, 15.91, 15.93, 15.94, 15.96, 15.97, 15.99, 16.00, 16.01, 16.03, 16.04, 16.06, 16.07, 16.09, 16.10, 16.12, 16.13, 16.15, 16.16, 16.18, 16.19, 16.21, 16.22, 16.24, 16.25, 16.27, 16.28, 16.30, 16.31, 16.33, 16.34, 16.35, 16.37, 16.38, 16.40, 16.41, 16.43, 16.44, 16.46, 16.47, 16.49, 16.50, 16.52, 16.53, 16.55, 16.56, 16.58, 16.59, 16.61, 16.62, 16.64, 16.66, 16.67, 16.69, 16.70, 16.72, 16.73, 16.75, 16.76, 16.78, 16.79, 16.81, 16.82, 16.84, 16.85, 16.87, 16.88, 16.90, 16.91, 16.93, 16.94, 16.96, 16.97, 16.99, 17.01, 17.02, 17.04, 17.05, 17.07, 17.08, 17.10, 17.11, 17.13, 17.14, 17.16, 17.17, 17.19, 17.21, 17.22, 17.24, 17.25, 17.27, 17.28, 17.30, 17.31, 17.33, 17.35, 17.36, 17.38, 17.39, 17.41, 17.42, 17.44, 17.45, 17.47, 17.49, 17.50, 17.52, 17.53, 17.55, 17.56, 17.58, 17.60, 17.61, 17.63, 17.64, 17.66, 17.67, 17.69, 17.71, 17.72, 17.74, 17.75, 17.77, 17.79, 17.80, 17.82, 17.83, 17.85, 17.86, 17.88, 17.90, 17.91, 17.93, 17.94, 17.96, 17.98, 17.99, 18.01, 18.02, 18.04, 18.06, 18.07, 18.09, 18.11, 18.12, 18.14, 18.15, 18.17, 18.19, 18.20, 18.22, 18.23, 18.25, 18.27, 18.28, 18.30, 18.32, 18.33, 18.35, 18.36, 18.38, 18.40, 18.41, 18.43, 18.45, 18.46, 18.48, 18.49, 18.51, 18.53, 18.54, 18.56, 18.58, 18.59, 18.61, 18.63, 18.64, 18.66, 18.68, 18.69, 18.71, 18.73, 18.74, 18.76, 18.77, 18.79, 18.81, 18.82, 18.84, 18.86, 18.87, 18.89, 18.91, 18.92, 18.94, 18.96, 18.97, 18.99, 19.01, 19.02, 19.04, 19.06, 19.08, 19.09, 19.10

11, 19.13, 19.14, 19.16, 19.18, 19.19, 19.21, 19.23, 19.24, 19.26, 19.28, 19.29, 19.31, 19.33, 19.35, 19.36, 19.38, 19.40, 19.41, 19.43, 19.45, 19.46, 19.48, 19.50, 19.52, 19.53, 19.55, 19.57, 19.58, 19.60, 19.62, 19.64, 19.65, 19.67, 19.69, 19.70, 19.72, 19.74, 19.76, 19.77, 19.79, 19.81, 19.83, 19.84, 19.86, 19.88, 19.90, 19.91, 19.93, 19.95, 19.97, 19.98, 20.00, 20.02, 20.04, 20.05, 20.07, 20.09, 20.11, 20.12, 20.14, 20.16, 20.18, 20.19, 20.21, 20.23, 20.25, 20.26, 20.28, 20.30, 20.32, 20.33, 20.35, 20.37, 20.39, 20.41, 20.42, 20.44, 20.46, 20.48, 20.49, 20.51, 20.53, 20.55, 20.57, 20.58, 20.60, 20.62, 20.64, 20.66, 20.67, 20.69, 20.71, 20.73, 20.75, 20.76, 20.78, 20.80, 20.82, 20.84, 20.85, 20.87, 20.89, 20.91, 20.93, 20.95, 20.96, 20.98, 21.00, 21.02, 21.04, 21.06, 21.07, 21.09, 21.11, 21.13, 21.15, 21.17, 21.18, 21.20, 21.22, 21.24, 21.26, 21.28, 21.29, 21.31, 21.33, 21.35, 21.37, 21.39, 21.41, 21.42, 21.44, 21.46, 21.48, 21.50, 21.52, 21.54, 21.55, 21.57, 21.59, 21.61, 21.63, 21.65, 21.67, 21.69, 21.70, 21.72, 21.74, 21.76, 21.78, 21.80, 21.82, 21.84, 21.86, 21.87, 21.89, 21.91, 21.93, 21.95, 21.97, 21.99, 22.01, 22.03, 22.05, 22.06, 22.08, 22.10, 22.12, 22.14, 22.16, 22.18, 22.20, 22.22, 22.24, 22.26, 22.28, 22.30, 22.31, 22.33, 22.35, 22.37, 22.39, 22.41, 22.43, 22.45, 22.47, 22.49, 22.51, 22.53, 22.55, 22.57, 22.59, 22.61, 22.63, 22.64, 22.66, 22.68, 22.70, 22.72, 22.74, 22.76, 22.78, 22.80, 22.82, 22.84, 22.86, 22.88, 22.90, 22.92, 22.94, 22.96, 22.98, 23.00, 23.02, 23.04, 23.06, 23.08, 23.10, 23.12, 23.14, 23.16, 23.18, 23.20, 23.22, 23.24, 23.26, 23.28, 23.30, 23.32, 23.34, 23.36, 23.38, 23.40, 23.42, 23.44, 23.46, 23.48, 23.50, 23.52, 23.54, 23.56, 23.58, 23.60, 23.62, 23.65, 23.67, 23.69, 23.71, 23.73, 23.75, 23.77, 23.79, 23.81, 23.83, 23.85, 23.87, 23.89, 23.91, 23.93, 23.95, 23.97, 24.00, 24.02, 24.04, 24.06, 24.08, 24.10, 24.12, 24.14, 24.16, 24.18, 24.20, 24.22, 24.25, 24.27, 24.29, 24.31, 24.33, 24.35, 24.37, 24.39, 24.41, 24.43, 24.46, 24.48, 24.50, 24.52, 24.54, 24.56, 24.58, 24.60, 24.63, 24.65, 24.67, 24.69, 24.71, 24.73, 24.75, 24.78, 24.80, 24.82, 24.84, 24.86, 24.88, 24.90, 24.93, 24.95, 24.97, 24.99, 25.01, 25.03, 25.06, 25.08, 25.10, 25.12, 25.14, 25.16, 25.19, 25.21, 25.23, 25.25, 25.27, 25.30, 25.32, 25.34, 25.36, 25.38, 25.41, 25.43, 25.45, 25.47, 25.49, 25.52, 25.54, 25.56, 25.58, 25.60, 25.63, 25.65, 25.67, 25.69, 25.72, 25.74, 25.76, 25.78, 25.81, 25.83, 25.85, 25.87, 25.89, 25.92, 25.94, 25.96, 25.98, 26.01, 26.03, 26.05, 26.08, 26.10, 26.12, 26.14, 26.17, 26.19, 26.21, 26.23, 26.26, 26.28, 26.30, 26.33, 26.35, 26.37, 26.39, 26.42, 26.44, 26.46, 26.49, 26.51, 26.53, 26.56, 26.58, 26.60, 26.63, 26.65, 26.67, 26.69, 26.72, 26.74, 26.76, 26.79, 26.81, 26.83, 26.86, 26.88, 26.90, 26.93, 26.95, 26.98, 27.00, 27.02, 27.05, 27.07, 27.09, 27.12, 27.14, 27.16, 27.19, 27.21, 27.24, 27.26, 27.28, 27.31, 27.33, 27.35, 27.38, 27.40, 27.43, 27.45, 27.47, 27.50, 27.52, 27.55, 27.57, 27.59, 27.62, 27.64, 27.67, 27.69, 27.72, 27.74, 27.76, 27.79, 27.81, 27.84, 27.86, 27.89, 27.91, 27.93, 27.96, 27.98, 28.01, 28.03, 28.06, 28.08, 28.11, 28.13, 28.16, 28.18, 28.21, 28.23, 28.26, 28.28, 28.30, 28.33, 28.35, 28.38, 28.40, 28.43, 28.45, 28.48, 28.50, 28.53, 28.55, 28.58, 28.60, 28.63, 28.66, 28.68, 28.71, 28.73, 28.76, 28.78, 28.81, 28.83, 28.86, 28.88, 28.91, 28.93, 28.96, 28.99, 29.01, 29.04, 29.06, 29.09, 29.11, 29.14, 29.17, 29.19, 29.22, 29.24, 29.27, 29.29, 29.32, 29.35, 29.37, 29.40, 29.42, 29.45, 29.48, 29.50, 29.53, 29.55, 29.58, 29.61, 29.63, 29.66, 29.69, 29.71, 29.74, 29.76, 29.79, 29.82, 29.84, 29.87, 29.90, 29.92, 29.95, 29.98, 30.00, 30.03, 30.06, 30.08, 30.11, 30.14, 30.16, 30.19, 30.22, 30.24, 30.27, 30.30, 30.33, 30.35, 30.38, 30.41, 30.43, 30.46, 30.49, 30.52, 30.54, 30.57, 30.60, 30.62, 30.65, 30.68, 30.71, 30.73, 30.76, 30.79, 30.82, 30.84, 30.87, 30.90, 30.93, 30.96, 30.98, 31.01, 31.04, 31.07, 31.09, 31.12, 31.15, 31.18, 31.21, 31.23, 31.26, 31.29, 31.32, 31.35, 31.37, 31.40, 31.43, 31.46, 31.49, 31.52, 31.54, 31.57, 31.60, 31.63, 31.66, 31.69, 31.72, 31.74, 31.77, 31.80, 31.83, 31.86, 31.89, 31.92, 31.95, 31.97, 32.00, 32.03, 32.06, 32.09, 32.12, 32.15, 32.18, 32.21, 32.24, 32.27, 32.29, 32.32, 32.35, 32.38, 32.41, 32.44, 32.47, 32.50, 32.53, 32.56, 32.59, 32.62, 32.65, 32.68, 32.71, 32.74, 32.77, 32.80, 32.83, 32.86, 32.89, 32.92, 32.95, 32.98, 33.01, 33.04, 33.07, 33.10, 33.13, 33.16, 33.19, 33.22, 33.25, 33.28, 33.31, 33.34, 33.37, 33.40, 33.43, 33.46, 33.49, 33.53, 33.56, 33.59, 33.62, 33.65, 33.68, 33.71, 33.74, 33.77, 33.80, 33.84, 33.87, 33.90, 33.93, 33.96, 33.99, 34.02, 34.05, 34.09, 34.12, 34.15, 34.18, 34.21, 34.24, 34.28, 34.31, 34.34, 34.37, 34.40, 34.43, 34.47, 34.50, 34.53, 34.56, 34.59, 34.63, 34.66, 34.69, 34.72, 34.76, 34.79, 34.82, 34.85, 34.89, 34.92, 34.95, 34.98, 35.02, 35.05, 35.08, 35.11, 35.15, 35.18, 35.21, 35.25, 35.28, 35.31, 35.35, 35.38, 35.41, 35.44, 35.48, 35.51, 35.54, 35.58, 35.61, 35.65, 35.68, 35.71, 35.75, 35.78, 35.81, 35.85, 35.88, 35.91, 35.95, 35.98, 36.02, 36.05, 36.08, 36.12, 36.15, 36.19, 36.22, 36.26, 36.29, 36.33, 36.36, 36.39, 36.43, 36.46, 36.50, 36.53, 36.57, 36.60, 36.64, 36.67, 36.71, 36.74, 36.78, 36.81, 36.85, 36.88, 36.92, 36.95, 36.99, 37.02, 37.06, 37.09, 37.13, 37.17, 37.20, 37.24, 37.27, 37.31, 37.34, 37.38, 37.42, 37.45, 37.49, 37.52, 37.56, 37.60, 37.63, 37.67, 37.71, 37.74, 37.78, 37.82, 37.85, 37.89, 37.93, 37.96, 38.00, 38.04, 38.07, 38.11, 38.15, 38.18, 38.22, 38.26, 38.30, 38.33, 38.37, 38.41, 38.44, 38.48, 38.52, 38.56, 38.60, 38.63, 38.67, 38.71, 38.75, 38.78, 38.82, 38.86, 38.90, 38.94, 38.97, 39.01, 39.05, 39.09, 39.13, 39.17, 39.21, 39.24, 39.28, 39.32, 39.36, 39.40, 39.44, 39.48, 39.52, 39.56, 39.59, 39.63, 39.67, 39.71, 39.75, 39.79, 39.83, 39.87, 39.91, 39.95, 39.99, 40.03, 40.07, 40.11, 40.15, 40.19, 40.23, 40.27, 40.31, 40.35, 40.39, 40.43, 40.47, 40.51, 40.55, 40.59, 40.64, 40.68, 40.72, 40.76, 40.80, 40.84, 40.88, 40.92, 40.96, 41.01, 41.05, 41.09, 41.13, 41.17, 41.21, 41.26, 41.30, 41.34, 41.38, 41.42, 41.47, 41.51, 41.55, 41.59, 41.64, 41.68, 41.72, 41.76, 41.81, 41.85, 41.89, 41.93, 41.98, 42.02, 42.06, 42.11, 42.15, 42.19, 42.24, 42.28, 42.32, 42.37, 42.41, 42.46, 42.50, 42.54, 42.59, 42.63, 42.68, 42.72, 42.76, 42.81, 42.85, 42.90, 42.94, 42.99, 43.03, 43.07, 43.11, 43.15, 43.19, 43.23, 43.27, 43.31, 43.35, 43.39, 43.43, 43.47, 43.51, 43.55, 43.59, 43.63, 43.67, 43.71, 43.75, 43.79, 43.83, 43.87, 43.91, 43.95, 43.99, 44.03, 44.07, 44.11, 44.15, 44.19, 44.23, 44.27, 44.31, 44.35, 44.39, 44.43, 44.47, 44.51, 44.55, 44.59, 44.63, 44.67, 44.71, 44.75, 44.79, 44.83, 44.87, 44.91, 44.95, 44.99, 45.03, 45.07, 45.11, 45.15, 45.19, 45.23, 45.27, 45.31, 45.35, 45.39, 45.43, 45.47, 45.51, 45.55, 45.59, 45.63, 45.67, 45.71, 45.75, 45.79, 45.83, 45.87, 45.91, 45.95, 45.99, 46.03, 46.07, 46.11, 46.15, 46.19, 46.23, 46.27, 46.31, 46.35, 46.39, 46.43, 46.47, 46.51, 46.55, 46.59, 46.63, 46.67, 46.71, 46.75, 46.79, 46.83, 46.87, 46.91, 46.95, 46.99, 47.03, 47.07, 47.11, 47.15, 47.19, 47.23, 47.27, 47.31, 47.35, 47.39, 47.43, 47.47, 47.51, 47.55, 47.59, 47.63, 47.67, 47.71, 47.75, 47.79, 47.83, 47.87, 47.91, 47.95, 47.99, 48.03, 48.07, 48.11, 48.15, 48.19, 48.23, 48.27, 48.31, 48.35, 48.39, 48.43, 48.47, 48.51, 48.55, 48.59, 48.63, 48.67, 48.71, 48.75, 48.79, 48.83, 48.87, 48.91, 48.95, 48.99, 49.03, 49.07, 49.11, 49.15, 49.19, 49.23, 49.27, 49.31, 49.35, 49.39, 49.43, 49.47, 49.51, 49.55, 49.59, 49.63, 49.67, 49.71, 49.75, 49.79, 49.83, 49.87, 49.91, 49.95, 49.99, 50.03, 50.07, 50.11, 50.15, 50.19, 50.23, 50.27, 50.31, 50.35, 50.39, 50.43, 50.47, 50.51, 50.55, 50.59, 50.63, 50.67, 50.71, 50.75, 50.79, 50.83, 50.87, 50.91, 50.95, 50.99, 51.03, 51.07, 51.11, 51.15, 51.19, 51.23, 51.27, 51.31, 51.35, 51.39, 51.43, 51.47, 51.51, 51.55, 51.59, 51.63, 51.67, 51.71, 51.75, 51.79, 51.83, 51.87, 51.91, 51.95, 51.99, 52.03, 52.07, 52.11, 52.15, 52.19, 52.23, 52.27, 52.31, 52.35, 52.39, 52.43, 52.47, 52.51, 52.55, 52.59, 52.63, 52.67, 52.71, 52.75, 52.79, 52.83, 52.87, 52.91, 52.95, 52.99, 53.03, 53.07, 53.11, 53.15, 53.19, 53.23, 53.27, 53.31, 53.35, 53.39, 53.43, 53.47, 53.51, 53.55, 53.59, 53.63, 53.67, 53.71, 53.75, 53.79, 53.83, 53.87, 53.91, 53.95, 53.99, 54.03, 54.07, 54.11, 54.15, 54.19, 54.23, 54.27, 54.31, 54.35, 54.39, 54.43, 54.47, 54.51, 54.55, 54.59, 54.63, 54.67, 54.71, 54.75, 54.79, 54.83, 54.87, 54.91, 54.95, 54.99, 55.03, 55.07, 55.11, 55.15, 55.19, 55.23, 55.27, 55.31, 55.35, 55.39, 55.43, 55.47, 55.51, 55.55, 55.59, 55.63, 55.67, 55.71, 55.75, 55.79, 55.83, 55.87, 55.91, 55.95, 55.99, 56.03, 56.07, 56.11, 56.15, 56.19, 56.23, 56.27, 56.31, 56.35, 56.39, 56.43, 56.47, 56.51, 56.55, 56.59, 56.63, 56.67, 56.71, 56.75, 56.79, 56.83, 56.87, 56.91, 56.95, 56.99, 57.03, 57.07, 57.11, 57.15, 57.19, 57.23, 57.27, 57.31, 57.35, 57.39, 57.43, 57.47, 57.51, 57.55, 57.59, 57.63, 57.67, 57.71, 57.75, 57.79, 57.83, 57.87, 57.91, 57.95, 57.99, 58.03, 58.07, 58.11, 58.15, 58.19, 58.23, 58.27, 58.31, 58.35, 58.39, 58.43, 58.47, 58.51, 58.55, 58.59, 58.63, 58.67, 58.71, 58.75, 58.79, 58.83, 58.87, 58.91, 58.95, 58.99, 59.03, 59.07, 59.11, 59.15, 59.19, 59.23, 59.27, 59.31, 59.35, 59.39, 59.43, 59.47, 59.51, 59.55, 59.59, 59.63, 59.67, 59.71, 59.75, 59.79, 59.83, 59.87, 59.91, 59.95, 59.99, 60.03, 60.07, 60.11, 60.15, 60.19, 60.23, 60.27, 60.31, 60.35, 60.39, 60.43, 60.47, 60.51, 60.55, 60.59, 60.63, 60.67, 60.71, 60.75, 60.79, 60.83, 60.87, 60.91, 60.95, 60.99, 61.03, 61.07, 61.11, 61.15, 61.19, 61.23, 61.27, 61.31, 61.35, 61.39, 61.43, 61.47, 61.51, 61.55, 61.59, 61.63, 61.67, 61.71, 61.75, 61.79, 61.83, 61.87, 61.91, 61.95, 61.99, 62.03, 62.07, 62.11, 62.15, 62.19, 62.23, 62.27, 62.31, 62.35, 62.39, 62.43, 62.47, 62.51, 62.55, 62.59, 62.63, 62.67, 62.71, 62.75, 62.79, 62.83, 62.87, 62.91, 62.95, 62.99, 63.03, 63.07, 63.11, 63.15, 63.19, 63.23, 63.27, 63.31, 63.35, 63.39, 63.43, 63.47, 63.51, 63.55, 63.59, 63.63, 63.67, 63.71, 63.75, 63.79, 63.83, 63.87, 63.91, 63.95, 63.99, 64.03, 64.07, 64.11, 64.15, 64.19, 64.23, 64.27, 64.31, 64.35, 64.39, 64.43, 64.47, 64.51, 64.55, 64.59, 64.63, 64.67, 64.71, 64.75, 64.79, 64.83, 64.87, 64.91, 64.95, 64.99, 65.03, 65.07, 65.11, 65.15, 65.19, 65.23, 65.27, 65.31, 65.35, 65.39, 65.43, 65.47, 65.51, 65.55, 65.59, 65.63, 65.67, 65.71, 65.75, 65.79, 65.83, 65.87, 65.91, 65.95, 65.99, 66.03, 66.07, 66.11, 66.15, 66.19, 66.23, 66.27, 66.31, 66.35, 66.39, 66.43, 66.47, 66.51, 66.55, 66.59, 66.63, 66.67, 66.71, 66.75, 66.79, 66.83, 66.87, 66.91, 66.95, 66.99, 67.03, 67.07, 67.11, 67.15, 67.19, 67.23, 67.27, 67.31, 67.35, 67.39, 67.43, 67.47, 67.51, 67.55, 67.59, 67.63, 67.67, 67.71, 67.75, 67.79, 67.83, 67.87, 67.91, 67.95, 67.99, 68.03, 68.07, 68.11, 68.15, 68.19, 68.23, 68.27, 68.31, 68.35, 68.39, 68.43, 68.47, 68.51, 68.55, 68.59, 68.63, 68.67, 68.71, 68.75, 68.79, 68.83, 68.87, 68.91, 68.95, 68.99, 69.03, 69.07, 69.11, 69.15, 69.19, 69.23, 69.27, 69.31, 69.35, 69.39, 69.43, 69.47, 69.51, 69.55, 69.59, 69.63, 69.67, 69.71, 69.75, 69.79, 69.83, 69.87, 69.91, 69.95, 69.99, 70.03, 70.07, 70.11, 70.15, 70.19, 70.23, 70.27, 70.31, 70.35, 70.39, 70.43, 70.47, 70.51, 70.55, 70.59, 70.63, 70.67, 70.71, 70.75, 70.79, 70.83, 70.87, 70.91, 70.95, 70.99, 71.03, 71.07, 71.11, 71.15, 71.19, 71.23, 71.27, 71.31, 71.35, 71.39, 71.43, 71.47, 71.51, 71.55, 71.59, 71.63, 71.67, 71.71, 71.75, 71.79, 71.83, 71.87, 71.91, 71.95, 71.99, 72.03, 72.07, 72.11, 72.15, 72.19, 72.23, 72.27, 72.31, 72.35, 72.39, 72.43, 72.47, 72.51, 72.55, 72.59, 72.63, 72.67, 72.71, 72.75, 72.79, 72.83, 72.87, 72.91, 72.95, 72.99, 73.03, 73.07, 73.11, 73.15, 73.19, 73.23, 73.27, 73.31, 73.35, 73.39, 73.43, 73.47, 73.51, 73.55, 73.59, 73.63, 73.67, 73.71, 73.75, 73.79, 73.83, 73.87, 73.91, 73.95, 73.99, 74.03, 74.07, 74.11, 74.15, 74.19, 74.23, 74.27, 74.31, 74.35, 74.39, 74.43, 74.47, 74.51, 74.55, 74.59, 74.63, 74.67, 74.71, 74.75, 74.79, 74.83, 74.87, 74.91, 74.95, 74.99, 75.03, 75.07, 75.11, 75.15, 75.19, 75.23, 75.27, 75.31, 75.35, 75.39, 75.43, 75.47, 75.51, 75.55, 75.59, 75.63, 75.67, 75.71, 75.75, 75.79, 75.83, 75.87, 75.91, 75.95, 75.99, 76.03, 76.07, 76.11, 76.15, 76.19, 76.23, 76.27, 76.31, 76.35, 76.39, 76.43, 76.47, 76.51, 76.55, 76.59, 76.63, 76.67, 76.71, 76.75, 76.79, 76.83, 76.87, 76.91, 76.95, 76.99, 77.03, 77.07, 77.11, 77.15, 77.19, 77.23, 77.27, 77.31, 77.35

Generated by Doxygen



2.98, 2.99, 3.00, 3.01, 3.02, 3.02, 3.03, 3.04, 3.05, 3.06, 3.07, 3.07, 3.08, 3.09, 3.10, 3.11, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.16, 3.17, 3.18, 3.19, 3.20, 3.20, 3.21, 3.22, 3.23, 3.24, 3.24, 3.25, 3.26, 3.27, 3.28, 3.29, 3.29, 3.30, 3.31, 3.32, 3.33, 3.34, 3.34, 3.35, 3.36, 3.37, 3.38, 3.38, 3.39, 3.40, 3.41, 3.42, 3.43, 3.43, 3.44, 3.45, 3.46, 3.47, 3.48, 3.48, 3.49, 3.50, 3.51, 3.52, 3.53, 3.53, 3.54, 3.55, 3.56, 3.57, 3.58, 3.58, 3.59, 3.60, 3.61, 3.62, 3.63, 3.63, 3.64, 3.65, 3.66, 3.67, 3.68, 3.68, 3.69, 3.70, 3.71, 3.72, 3.73, 3.73, 3.74, 3.75, 3.76, 3.77, 3.78, 3.78, 3.79, 3.80, 3.81, 3.82, 3.83, 3.83, 3.84, 3.85, 3.86, 3.87, 3.88, 3.89, 3.89, 3.90, 3.91, 3.92, 3.93, 3.94, 3.94, 3.95, 3.96, 3.97, 3.98, 3.99, 4.00, 4.00, 4.01, 4.02, 4.03, 4.04, 4.05, 4.05, 4.06, 4.07, 4.08, 4.09, 4.10, 4.11, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.17, 4.18, 4.19, 4.20, 4.21, 4.22, 4.23, 4.23, 4.24, 4.25, 4.26, 4.27, 4.28, 4.29, 4.29, 4.30, 4.31, 4.32, 4.33, 4.34, 4.35, 4.35, 4.36, 4.37, 4.38, 4.39, 4.40, 4.41, 4.42, 4.42, 4.43, 4.44, 4.45, 4.46, 4.47, 4.48, 4.48, 4.49, 4.50, 4.51, 4.52, 4.53, 4.54, 4.55, 4.55, 4.56, 4.57, 4.58, 4.59, 4.60, 4.61, 4.62, 4.62, 4.63, 4.64, 4.65, 4.66, 4.67, 4.68, 4.69, 4.69, 4.70, 4.71, 4.72, 4.73, 4.74, 4.75, 4.76, 4.76, 4.77, 4.78, 4.79, 4.80, 4.81, 4.82, 4.83, 4.83, 4.84, 4.85, 4.86, 4.87, 4.88, 4.89, 4.90, 4.91, 4.91, 4.92, 4.93, 4.94, 4.95, 4.96, 4.97, 4.98, 4.99, 4.99, 5.00, 5.01, 5.02, 5.03, 5.04, 5.05, 5.06, 5.07, 5.07, 5.08, 5.09, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.16, 5.17, 5.18, 5.19, 5.20, 5.21, 5.22, 5.23, 5.24, 5.24, 5.25, 5.26, 5.27, 5.28, 5.29, 5.30, 5.31, 5.32, 5.33, 5.34, 5.34, 5.35, 5.36, 5.37, 5.38, 5.39, 5.40, 5.41, 5.42, 5.43, 5.43, 5.44, 5.45, 5.46, 5.47, 5.48, 5.49, 5.50, 5.51, 5.52, 5.53, 5.53, 5.54, 5.55, 5.56, 5.57, 5.58, 5.59, 5.60, 5.61, 5.62, 5.63, 5.64, 5.64, 5.65, 5.66, 5.67, 5.68, 5.69, 5.70, 5.71, 5.72, 5.73, 5.74, 5.75, 5.75, 5.76, 5.77, 5.78, 5.79, 5.80, 5.81, 5.82, 5.83, 5.84, 5.85, 5.86, 5.87, 5.88, 5.88, 5.89, 5.90, 5.91, 5.92, 5.93, 5.94, 5.95, 5.96, 5.97, 5.98, 5.99, 6.00, 6.01, 6.01, 6.02, 6.03, 6.04, 6.05, 6.06, 6.07, 6.08, 6.09, 6.10, 6.11, 6.12, 6.13, 6.14, 6.15, 6.16, 6.16, 6.17, 6.18, 6.19, 6.20, 6.21, 6.22, 6.23, 6.24, 6.25, 6.26, 6.27, 6.28, 6.29, 6.30, 6.31, 6.32, 6.32, 6.33, 6.34, 6.35, 6.36, 6.37, 6.38, 6.39, 6.40, 6.41, 6.42, 6.43, 6.44, 6.45, 6.46, 6.47, 6.48, 6.49, 6.50, 6.51, 6.51, 6.52, 6.53, 6.54, 6.55, 6.56, 6.57, 6.58, 6.59, 6.60, 6.61, 6.62, 6.63, 6.64, 6.65, 6.66, 6.67, 6.68, 6.69, 6.70, 6.71, 6.72, 6.73, 6.74, 6.75, 6.75, 6.76, 6.77, 6.78, 6.79, 6.80, 6.81, 6.82, 6.83, 6.84, 6.85, 6.86, 6.87, 6.88, 6.89, 6.90, 6.91, 6.92, 6.93, 6.94, 6.95, 6.96, 6.97, 6.98, 6.99, 7.00, 7.01, 7.02, 7.03, 7.04, 7.05, 7.06, 7.07, 7.08, 7.09, 7.10, 7.11, 7.12, 7.12, 7.13, 7.14, 7.15, 7.16, 7.17, 7.18, 7.19, 7.20, 7.21, 7.22, 7.23, 7.24, 7.25, 7.26, 7.27, 7.28, 7.29, 7.30, 7.31, 7.32, 7.33, 7.34, 7.35, 7.36, 7.37, 7.38, 7.39, 7.40, 7.41, 7.42, 7.43, 7.44, 7.45, 7.46, 7.47, 7.48, 7.49, 7.50, 7.51, 7.52, 7.53, 7.54, 7.55, 7.56, 7.57, 7.58, 7.59, 7.60, 7.61, 7.62, 7.63, 7.64, 7.65, 7.66, 7.67, 7.68, 7.69, 7.70, 7.71, 7.72, 7.73, 7.74, 7.75, 7.76, 7.77, 7.78, 7.79, 7.80, 7.81, 7.82, 7.83, 7.84, 7.85, 7.86, 7.87, 7.88, 7.89, 7.91, 7.92, 7.93, 7.94, 7.95, 7.96, 7.97, 7.98, 7.99, 8.00, 8.01, 8.02, 8.03, 8.04, 8.05, 8.06, 8.07, 8.08, 8.09, 8.10, 8.11, 8.12, 8.13, 8.14, 8.15, 8.16, 8.17, 8.18, 8.19, 8.20, 8.21, 8.22, 8.23, 8.24, 8.25, 8.26, 8.27, 8.29, 8.30, 8.31, 8.32, 8.33, 8.34, 8.35, 8.36, 8.37, 8.38, 8.39, 8.40, 8.41, 8.42, 8.43, 8.44, 8.45, 8.46, 8.47, 8.48, 8.49, 8.50, 8.51, 8.52, 8.54, 8.55, 8.56, 8.57, 8.58, 8.59, 8.60, 8.61, 8.62, 8.63, 8.64, 8.65, 8.66, 8.67, 8.68, 8.69, 8.70, 8.71, 8.72, 8.74, 8.75, 8.76, 8.77, 8.78, 8.79, 8.80, 8.81, 8.82, 8.83, 8.84, 8.85, 8.86, 8.87, 8.88, 8.89, 8.91, 8.92, 8.93, 8.94, 8.95, 8.96, 8.97, 8.98, 8.99, 9.00, 9.01, 9.02, 9.03, 9.04, 9.06, 9.07, 9.08, 9.09, 9.10, 9.11, 9.12, 9.13, 9.14, 9.15, 9.16, 9.17, 9.18, 9.20, 9.21, 9.22, 9.23, 9.24, 9.25, 9.26, 9.27, 9.28, 9.29, 9.30, 9.31, 9.33, 9.34, 9.35, 9.36, 9.37, 9.38, 9.39, 9.40, 9.41, 9.42, 9.43, 9.45, 9.46, 9.47, 9.48, 9.49, 9.50, 9.51, 9.52, 9.53, 9.54, 9.55, 9.57, 9.58, 9.59, 9.60, 9.61, 9.62, 9.63, 9.64, 9.65, 9.66, 9.68, 9.69, 9.70, 9.71, 9.72, 9.73, 9.74, 9.75, 9.76, 9.78, 9.79, 9.80, 9.81, 9.82, 9.83, 9.84, 9.85, 9.86, 9.88, 9.89, 9.90, 9.91, 9.92, 9.93, 9.94, 9.95, 9.96, 9.98, 9.99, 10.00, 10.01, 10.02, 10.03, 10.04, 10.05, 10.07, 10.08, 10.09, 10.10, 10.11, 10.12, 10.13, 10.14, 10.16, 10.17, 10.18, 10.19, 10.20, 10.21, 10.22, 10.24, 10.25, 10.26, 10.27, 10.28, 10.29, 10.30, 10.31, 10.33, 10.34, 10.35, 10.36, 10.37, 10.38, 10.39, 10.41, 10.42, 10.43, 10.44, 10.45, 10.46, 10.47, 10.49, 10.50, 10.51, 10.52, 10.53, 10.54, 10.55, 10.57, 10.58, 10.59, 10.60, 10.61, 10.62, 10.64, 10.65, 10.66, 10.67, 10.68, 10.69, 10.70, 10.72, 10.73, 10.74, 10.75, 10.76, 10.77, 10.79, 10.80, 10.81, 10.82, 10.83, 10.84, 10.86, 10.87, 10.88, 10.89, 10.90, 10.91, 10.93, 10.94, 10.95, 10.96, 10.97, 10.98, 11.00, 11.01, 11.02, 11.03, 11.04, 11.05, 11.07, 11.08, 11.09, 11.10, 11.11, 11.13, 11.14, 11.15, 11.16, 11.17, 11.18, 11.20, 11.21, 11.22, 11.23, 11.24, 11.26, 11.27, 11.28, 11.29, 11.30, 11.32, 11.33, 11.34, 11.35, 11.36, 11.37, 11.39, 11.40, 11.41, 11.42, 11.43, 11.45, 11.46, 11.47, 11.48, 11.49, 11.51, 11.52, 11.53, 11.54, 11.55, 11.57, 11.58, 11.59, 11.60, 11.61, 11.63, 11.64, 11.65, 11.66, 11.68, 11.69, 11.70, 11.71, 11.72, 11.74, 11.75, 11.76, 11.77, 11.78, 11.80, 11.81, 11.82, 11.83, 11.85, 11.86, 11.87, 11.88, 11.89, 11.91, 11.92, 11.93, 11.94, 11.96, 11.97, 11.98, 11.99, 12.00, 12.02, 12.03, 12.04, 12.05, 12.07, 12.08, 12.09, 12.10, 12.11, 12.13, 12.14, 12.15, 12.16, 12.18, 12.19, 12.20, 12.21, 12.23, 12.24, 12.25, 12.26, 12.28, 12.29, 12.30, 12.31, 12.33, 12.34, 12.35, 12.36, 12.38, 12.39, 12.40, 12.41, 12.43, 12.44, 12.45, 12.46, 12.48, 12.49, 12.50, 12.51, 12.53, 12.54, 12.55, 12.56, 12.58, 12.59, 12.60, 12.61, 12.63, 12.64, 12.65, 12.66, 12.68, 12.69, 12.70, 12.72, 12.73, 12.74, 12.75, 12.77, 12.78, 12.79, 12.80, 12.82, 12.83, 12.84, 12.86, 12.87, 12.88, 12.89, 12.91, 12.92, 12.93, 12.94, 12.96, 12.97, 12.98, 13.00, 13.01, 13.02, 13.03, 13.05, 13.06, 13.07, 13.09, 13.10, 13.11, 13.12, 13.14, 13.15, 13.16, 13.18, 13.19, 13.20, 13.22, 13.23, 13.24, 13.25, 13.27, 13.28, 13.29, 13.31, 13.32, 13.33, 13.35, 13.36, 13.37, 13.38, 13.40, 13.41, 13.42, 13.44, 13.45, 13.46, 13.48, 13.49, 13.50, 13.52, 13.53, 13.54, 13.55, 13.57, 13.58, 13.59

Generated by Doxygen

27, 30.30, 30.33, 30.35, 30.38, 30.41, 30.43, 30.46, 30.49, 30.52, 30.54, 30.57, 30.60, 30.62, 30.65, 30.68, 30.71, 30.73, 30.76, 30.79, 30.82, 30.84, 30.87, 30.90, 30.93, 30.96, 30.98, 31.01, 31.04, 31.07, 31.09, 31.12, 31.15, 31.18, 31.21, 31.23, 31.26, 31.29, 31.32, 31.35, 31.37, 31.40, 31.43, 31.46, 31.49, 31.52, 31.54, 31.57, 31.60, 31.63, 31.66, 31.69, 31.72, 31.74, 31.77, 31.80, 31.83, 31.86, 31.89, 31.92, 31.95, 31.97, 32.00, 32.03, 32.06, 32.09, 32.12, 32.15, 32.18, 32.21, 32.24, 32.27, 32.29, 32.32, 32.35, 32.38, 32.41, 32.44, 32.47, 32.50, 32.53, 32.56, 32.59, 32.62, 32.65, 32.68, 32.71, 32.74, 32.77, 32.80, 32.83, 32.86, 32.89, 32.92, 32.95, 32.98, 33.01, 33.04, 33.07, 33.10, 33.13, 33.16, 33.19, 33.22, 33.25, 33.28, 33.31, 33.34, 33.37, 33.40, 33.43, 33.46, 33.49, 33.53, 33.56, 33.59, 33.62, 33.65, 33.68, 33.71, 33.74, 33.77, 33.80, 33.84, 33.87, 33.90, 33.93, 33.96, 33.99, 34.02, 34.05, 34.09, 34.12, 34.15, 34.18, 34.21, 34.24, 34.28, 34.31, 34.34, 34.37, 34.40, 34.43, 34.47, 34.50, 34.53, 34.56, 34.59, 34.63, 34.66, 34.69, 34.72, 34.76, 34.79, 34.82, 34.85, 34.89, 34.92, 34.95, 34.98, 35.02, 35.05, 35.08, 35.11, 35.15, 35.18, 35.21, 35.25, 35.28, 35.31, 35.35, 35.38, 35.41, 35.44, 35.48, 35.51, 35.54, 35.58, 35.61, 35.65, 35.68, 35.71, 35.75, 35.78, 35.81, 35.85, 35.88, 35.91, 35.95, 35.98, 36.02, 36.05, 36.08, 36.12, 36.15, 36.19, 36.22, 36.26, 36.29, 36.33, 36.36, 36.39, 36.43, 36.46, 36.50, 36.53, 36.57, 36.60, 36.64, 36.67, 36.71, 36.74, 36.78, 36.81, 36.85, 36.88, 36.92, 36.95, 36.99, 37.02, 37.06, 37.09, 37.13, 37.17, 37.20, 37.24, 37.27, 37.31, 37.34, 37.38, 37.42, 37.45, 37.49, 37.52, 37.56, 37.60, 37.63, 37.67, 37.71, 37.74, 37.78, 37.82, 37.85, 37.89, 37.93, 37.96, 38.00, 38.04, 38.07, 38.11, 38.15, 38.18, 38.22, 38.26, 38.30, 38.33, 38.37, 38.41, 38.44, 38.48, 38.52, 38.56, 38.60, 38.63, 38.67, 38.71, 38.75, 38.78, 38.82, 38.86, 38.90, 38.94, 38.97, 39.01, 39.05, 39.09, 39.13, 39.17, 39.21, 39.24, 39.28, 39.32, 39.36, 39.40, 39.44, 39.48, 39.52, 39.56, 39.59, 39.63, 39.67, 39.71, 39.75, 39.79, 39.83, 39.87, 39.91, 39.95, 39.99, 40.03, 40.07, 40.11, 40.15, 40.19, 40.23, 40.27, 40.31, 40.35, 40.39, 40.43, 40.47, 40.51, 40.55, 40.59, 40.64, 40.68, 40.72, 40.76, 40.80, 40.84, 40.88, 40.92, 40.96, 41.01, 41.05, 41.09, 41.13, 41.17, 41.21, 41.26, 41.30, 41.34, 41.38, 41.42, 41.47, 41.51, 41.55, 41.59, 41.64, 41.68, 41.72, 41.76, 41.81, 41.85, 41.89, 41.93, 41.98, 42.02, 42.06, 42.11, 42.15, 42.19, 42.24, 42.28, 42.32, 42.37, 42.41, 42.46, 42.50, 42.54, 42.59, 42.63, 42.68, 42.72, 42.76, 42.81, 42.85, 42.90, 42.94, 42.99, 43.03, 43.08, 43.12, 43.17, 43.21, 43.26, 43.30, 43.35, 43.39, 43.44, 43.48, 43.53, 43.58, 43.62, 43.67, 43.71, 43.76, 43.81, 43.85, 43.90, 43.94, 43.99, 44.04, 44.08, 44.13, 44.18, 44.23, 44.27, 44.32, 44.37, 44.41, 44.46, 44.51, 44.56, 44.60, 44.65, 44.70, 44.75, 44.80, 44.84, 44.89, 44.94, 44.99, 45.04, 45.08, 45.13, 45.18, 45.23, 45.28, 45.33, 45.38, 45.43, 45.48, 45.53, 45.57, 45.62, 45.67, 45.72, 45.77, 45.82, 45.87, 45.92, 45.97, 46.02, 46.07, 46.12, 46.17, 46.23, 46.28, 46.33, 46.38, 46.43, 46.48, 46.53, 46.58, 46.63, 46.69, 46.74, 46.79, 46.84, 46.89, 46.95, 47.00, 47.05, 47.10, 47.15, 47.21, 47.26, 47.31, 47.37, 47.42, 47.47, 47.52, 47.58, 47.63, 47.69, 47.74, 47.79, 47.85, 47.90, 47.95, 48.01, 48.06, 48.12, 48.17, 48.23, 48.28, 48.34, 48.39, 48.45, 48.50, 48.56, 48.61, 48.67, 48.72, 48.78, 48.84, 48.89, 48.95, 49.00, 49.06, 49.12, 49.17, 49.23, 49.29, 49.34, 49.40, 49.46, 49.52, 49.57, 49.63, 49.69, 49.75, 49.80, 49.86, 49.92, 49.98, 50.04, 50.10, 50.16, 50.21, 50.27, 50.33, 50.39, 50.45, 50.51, 50.57, 50.63, 50.69, 50.75, 50.81, 50.87, 50.93, 50.99, 51.05, 51.11, 51.17, 51.24, 51.30, 51.36, 51.42, 51.48, 51.54, 51.61, 51.67, 51.73, 51.79, 51.86, 51.92, 51.98, 52.04, 52.11, 52.17, 52.23, 52.30, 52.36, 52.43, 52.49, 52.55, 52.62, 52.68, 52.75, 52.81, 52.88, 52.94, 53.01, 53.07, 53.14, 53.21, 53.27, 53.34, 53.40, 53.47, 53.54, 53.60, 53.67, 53.74, 53.80, 53.87, 53.94, 54.01, 54.08, 54.14, 54.21, 54.28, 54.35, 54.42, 54.49, 54.56, 54.63, 54.70, 54.76, 54.83, 54.90, 54.98, 55.05, 55.12, 55.19, 55.26, 55.33, 55.40, 55.47, 55.54, 55.62, 55.69, 55.76, 55.83, 55.91, 55.98, 56.05, 56.12, 56.20, 56.27, 56.35, 56.42, 56.49, 56.57, 56.64, 56.72, 56.79, 56.87, 56.94, 57.02, 57.10, 57.17, 57.25, 57.32, 57.40, 57.48, 57.56, 57.63, 57.71, 57.79, 57.87, 57.94, 58.02, 58.10, 58.18, 58.26, 58.34, 58.42, 58.50, 58.58, 58.66, 58.74, 58.82, 58.90, 58.98, 59.06, 59.15, 59.23, 59.31, 59.39, 59.48, 59.56, 59.64, 59.72, 59.81, 59.89, 59.98, 60.06, 60.15, 60.23, 60.32, 60.40, 60.49, 60.57, 60.66, 60.75, 60.83, 60.92, 61.01, 61.10, 61.18, 61.27, 61.36, 61.45, 61.54, 61.63, 61.72, 61.81, 61.90, 61.99, 62.08, 62.17, 62.26, 62.35, 62.44, 62.54, 62.63, 62.72, 62.82, 62.91, 63.00, 63.10, 63.19, 63.29, 63.38, 63.48, 63.57, 63.67, 63.76, 63.86, 63.96, 64.06, 64.15, 64.25, 64.35, 64.45, 64.55, 64.65, 64.75, 64.85, 64.95, 65.05, 65.15, 65.25, 65.35, 65.46, 65.56, 65.66, 65.76, 65.87, 65.97, 66.08, 66.18, 66.29, 66.39, 66.50, 66.61, 66.71, 66.82, 66.93, 67.03, 67.14, 67.25, 67.36, 67.47, 67.58, 67.69, 67.80, 67.91, 68.03, 68.14, 68.25, 68.36, 68.48, 68.59, 68.71, 68.82, 68.94, 69.05, 69.17, 69.29, 69.40, 69.52, 69.64, 69.76, 69.88, 70.00, 70.12, 70.24, 70.36, 70.48, 70.60, 70.73, 70.85, 70.97, 71.10, 71.22, 71.35, 71.47, 71.60, 71.73, 71.86, 71.98, 72.11, 72.24, 72.37, 72.50, 72.63, 72.76, 72.90, 73.03, 73.16, 73.30, 73.43, 73.57, 73.70, 73.84, 73.98, 74.11, 74.25, 74.39, 74.53, 74.67, 74.81, 74.95, 75.10, 75.24, 75.38, 75.53, 75.67, 75.82, 75.97, 76.11, 76.26, 76.41, 76.56, 76.71, 76.86, 77.01, 77.17, 77.32, 77.47, 77.63, 77.78, 77.94, 78.10, 78.26, 78.42, 78.58, 78.74, 78.90, 79.06, 79.23, 79.39, 79.55, 79.72, 79.89, 80.06, 80.23, 80.40, 80.57, 80.74, 80.91, 81.08, 81.26, 81.44, 81.61, 81.79, 81.97, 82.15, 82.33, 82.51, 82.70, 82.88, 83.07, 83.25, 83.44, 83.63, 83.82, 84.01, 84.20, 84.40, 84.59, 84.79, 84.98, 85.18, 85.38, 85.58, 85.79, 85.99, 86.20, 86.40, 86.61, 86.82, 87.03, 87.24, 87.46, 87.67, 87.89, 88.11, 88.33, 88.55, 88.77, 89.00, 89.22, 89.45, 89.68, 89.91, 90.14, 90.38, 90.62, 90.85, 91.09, 91.34, 91.58, 91.82, 92.07, 92.32, 92.57, 92.83, 93.08, 93.34, 93.60, 93.86, 94.13, 94.39, 94.66, 94.93, 95.20, 95.48, 95.76, 96.04, 96.32, 96.61, 96.89, 97.18, 97.45, 97.73, 98.01, 98.29, 98.57, 98.85, 99.13, 99.41, 99.69, 99.97, 100.25, 100.53, 100.81, 101.09, 101.37, 101.65, 101.93, 102.21, 102.49, 102.77, 103.05, 103.33, 103.61, 103.89, 104.17, 104.45, 104.73, 105.01, 105.29, 105.57, 105.85, 106.13, 106.41, 106.69, 106.97, 107.25, 107.53, 107.81, 108.09, 108.37, 108.65, 108.93, 109.21, 109.49, 109.77, 110.05, 110.33, 110.61, 110.89, 111.17, 111.45, 111.73, 112.01, 112.29, 112.57, 112.85, 113.13, 113.41, 113.69, 113.97, 114.25, 114.53, 114.81, 115.09, 115.37, 115.65, 115.93, 116.21, 116.49, 116.77, 117.05, 117.33, 117.61, 117.89, 118.17, 118.45, 118.73, 119.01, 119.29, 119.57, 119.85, 120.13, 120.41, 120.69, 120.97, 121.25, 121.53, 121.81, 122.09, 122.37, 122.65, 122.93, 123.21, 123.49, 123.77, 124.05, 124.33, 124.61, 124.89, 125.17, 125.45, 125.73, 126.01, 126.29, 126.57, 126.85, 127.13, 127.41, 127.69, 127.97, 128.25, 128.53, 128.81, 129.09, 129.37, 129.65, 129.93, 130.21, 130.49, 130.77, 131.05, 131.33, 131.61, 131.89, 132.17, 132.45, 132.73, 133.01, 133.29, 133.57, 133.85, 134.13, 134.41, 134.69, 134.97, 135.25, 135.53, 135.81, 136.09, 136.37, 136.65, 136.93, 137.21, 137.49, 137.77, 138.05, 138.33, 138.61, 138.89, 139.17, 139.45, 139.73, 140.01, 140.29, 140.57, 140.85, 141.13, 141.41, 141.69, 141.97, 142.25, 142.53, 142.81, 143.09, 143.37, 143.65, 143.93, 144.21, 144.49, 144.77, 145.05, 145.33, 145.61, 145.89, 146.17, 146.45, 146.73, 147.01, 147.29, 147.57, 147.85, 148.13, 148.41, 148.69, 148.97, 149.25, 149.53, 149.81, 150.09, 150.37, 150.65, 150.93, 151.21, 151.49, 151.77, 152.05, 152.33, 152.61, 152.89, 153.17, 153.45, 153.73, 154.01, 154.29, 154.57, 154.85, 155.13, 155.41, 155.69, 155.97, 156.25, 156.53, 156.81, 157.09, 157.37, 157.65, 157.93, 158.21, 158.49, 158.77, 159.05, 159.33, 159.61, 159.89, 160.17, 160.45, 160.73, 161.01, 161.29, 161.57, 161.85, 162.13, 162.41, 162.69, 162.97, 163.25, 163.53, 163.81, 164.09, 164.37, 164.65, 164.93, 165.21, 165.49, 165.77, 166.05, 166.33, 166.61, 166.89, 167.17, 167.45, 167.73, 168.01, 168.29, 168.57, 168.85, 169.13, 169.41, 169.69, 169.97, 170.25, 170.53, 170.81, 171.09, 171.37, 171.65, 171.93, 172.21, 172.49, 172.77, 173.05, 173.33, 173.61, 173.89, 174.17, 174.45, 174.73, 175.01, 175.29, 175.57, 175.85, 176.13, 176.41, 176.69, 176.97, 177.25, 177.53, 177.81, 178.09, 178.37, 178.65, 178.93, 179.21, 179.49, 179.77, 180.05, 180.33, 180.61, 180.89, 181.17, 181.45, 181.73, 182.01, 182.29, 182.57, 182.85, 183.13, 183.41, 183.69, 183.97, 184.25, 184.53, 184.81, 185.09, 185.37, 185.65, 185.93, 186.21, 186.49, 186.77, 187.05, 187.33, 187.61, 187.89, 188.17, 188.45, 188.73, 189.01, 189.29, 189.57, 189.85, 190.13, 190.41, 190.69, 190.97, 191.25, 191.53, 191.81, 192.09, 192.37, 192.65, 192.93, 193.21, 193.49, 193.77, 194.05, 194.33, 194.61, 194.89, 195.17, 195.45, 195.73, 196.01, 196.29, 196.57, 196.85, 197.13, 197.41, 197.69, 197.97, 198.25, 198.53, 198.81, 199.09, 199.37, 199.65, 199.93, 200.21, 200.49, 200.77, 201.05, 201.33, 201.61, 201.89, 202.17, 202.45, 202.73, 203.01, 203.29, 203.57, 203.85, 204.13, 204.41, 204.69, 204.97, 205.25, 205.53, 205.81, 206.09, 206.37, 206.65, 206.93, 207.21, 207.49, 207.77, 208.05, 208.33, 208.61, 208.89, 209.17, 209.45, 209.73, 210.01, 210.29, 210.57, 210.85, 211.13, 211.41, 211.69, 211.97, 212.25, 212.53, 212.81, 213.09, 213.37, 213.65, 213.93, 214.21, 214.49, 214.77, 215.05, 215.33, 215.61, 215.89, 216.17, 216.45, 216.73, 217.01, 217.29, 217.57, 217.85, 218.13, 218.41, 218.69, 218.97, 219.25, 219.53, 219.81, 220.09, 220.37, 220.65, 220.93, 221.21, 221.49, 221.77, 222.05, 222.33, 222.61, 222.89, 223.17, 223.45, 223.73, 224.01, 224.29, 224.57, 224.85, 225.13, 225.41, 225.69, 225.97, 226.25, 226.53, 226.81, 227.09, 227.37, 227.65, 227.93, 228.21, 228.49, 228.77, 229.05, 229.33, 229.61, 229.89, 230.17, 230.45, 230.73, 231.01, 231.29, 231.57, 231.85, 232.13, 232.41, 232.69, 232.97, 233.25, 233.53, 233.81, 234.09, 234.37, 234.65, 234.93, 235.21, 235.49, 235.77, 236.05, 236.33, 236.61, 236.89, 237.17, 237.45, 237.73, 238.01, 238.29, 238.57, 238.85, 239.13, 239.41, 239.69, 239.97, 240.25, 240.53, 240.81, 241.09, 241.37, 241.65, 241.93, 242.21, 242.49, 242.77, 243.05, 243.33, 243.61, 243.89, 244.17, 244.45, 244.73, 245.01, 245.29, 245.57, 245.85, 246.13, 246.41, 246.69, 246.97, 247.25, 247.53, 247.81, 248.09, 248.37, 248.65, 248.93, 249.21, 249.49, 249.77, 250.05, 250.33, 250.61, 250.89, 251.17, 251.45, 251.73, 252.01, 252.29, 252.57, 252.85, 253.13, 253.41, 253.69, 253.97, 254.25, 254.53, 254.81, 255.09, 255.37, 255.65, 255.93, 256.21, 256.49, 256.77, 257.05, 257.33, 257.61, 257.89, 258.17, 258.45, 258.73, 259.01, 259.29, 259.57, 259.85, 260.13, 260.41, 260.69, 260.97, 261.25, 261.53, 261.81, 262.09, 262.37, 262.65, 262.93, 263.21, 263.49, 263.77, 264.05, 264.33, 264.61, 264.89, 265.17, 265.45, 265.73, 266.01, 266.29, 266.57, 266.85, 267.13, 267.41, 267.69, 267.97, 268.25, 268.53, 268.81, 269.09, 269.37, 269.65, 269.93, 270.21, 270.49, 270.77, 271.05, 271.33, 271.61, 271.89, 272.17, 272.45, 272.73, 273.01, 273.29, 273.57, 273.85, 274.13, 274.41, 274.69, 274.97, 275.25, 275.53, 275.81, 276.09, 276.37, 276.65, 276.93, 277.21, 277.49, 277.77, 278.05, 278.33, 278.61, 278.89, 279.17, 279.45, 279.73, 280.01, 280.29, 280.57, 280.85, 281.13, 281.41, 281.69, 281.97, 282.25, 282.53, 282.81, 283.09, 283.37, 283.65, 283.93, 284.21, 284.49, 284.77, 285.05, 285.33, 285.61, 285.89, 286.17, 286.45, 286.73, 287.01, 287.29, 287.57, 287.85, 288.13, 288.41, 288.69, 288.97, 289.25, 289.53, 289.81, 290.09, 290.37, 290.65, 290.93, 291.21, 291.49, 291.77, 292.05, 292.33, 292.61, 292.89, 293.17, 293.45, 293.73, 294.01, 294.29, 294.57, 294.85, 295.13, 295.41, 295.69, 295.97, 296.25, 296.53, 296.81, 297.09, 297.37, 297.65, 297.93, 298.21, 298.49, 298.77, 299.05, 299.33, 299.61, 299.89, 300.17, 300.45, 300.73, 301.01, 301.29, 301.57, 301.85, 302.13, 302.41, 302.69, 302.97, 303.25, 303.53, 303.81, 304.09, 304.37, 304.65, 304.93, 305.21, 305.49, 305.77, 306.05, 306.33, 306.61, 30





## 4.33.2 Function Documentation

### 4.33.2.1 `calibrate_offsets()`

```
void calibrate_offsets (
    volatile uint16_t rawADC[],
    volatile float currentOffsets[],
    uint32_t numSamples )
```

Calibrate the current sensor offsets.

This function calculates the average offset for each current sensor channel by reading the ADC values when no current is flowing. The calculated offsets are used to correct the sensor readings.

#### Parameters

in	<i>rawADC</i>	Buffer containing the raw ADC values for the channels.
out	<i>currentOffsets</i>	Array to store the calculated offsets for each current channel.
in	<i>numSamples</i>	Number of samples to average for the offset calculation.

Here is the caller graph for this function:



### 4.33.2.2 `get_currents_voltage()`

```
uint8_t get_currents_voltage (
    volatile uint16_t ADC_raw[],
    volatile Analog * analog,
    volatile Feedback * feedback,
    volatile InverterError * errors,
    float sinTheta_e,
    float cosTheta_e )
```

Get electrical ADC measurements.

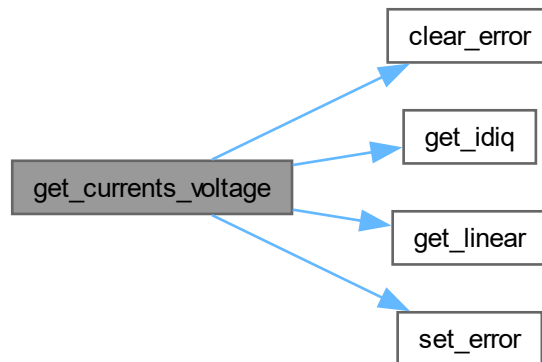
#### Parameters

in	<i>ADC_raw</i>	Pointer to the raw ADC values array.
out	<i>analog</i>	Pointer to the ADC struct to store the results.
out	<i>feedback</i>	Pointer to the <code>Feedback</code> struct to store id and iq.
in	<i>sinTheta_e</i>	Electrical angle sine (-1..1)
in	<i>cosTheta_e</i>	Electrical angle cosine (-1..1)

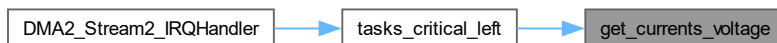
## Return values

<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:



#### 4.33.2.3 get\_idiq()

```

void get_idiq (
    float ia,
    float ib,
    float ic,
    float sinTheta_e,
    float cosTheta_e,
    float * idMeas,
    float * iqMeas )
  
```

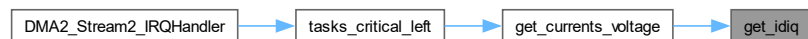
Computes d-q currents from current measurements and electrical angle.

This function computes the d-q currents from phase currents (ABC), `theta_e`, and stores the results in the provided pointers.

## Parameters

in	<i>ia</i>	Phase A current in A.
in	<i>ib</i>	Phase B current in A.
in	<i>ic</i>	Phase C current in A.
in	<i>sinTheta</i> <sub>e</sub>	Electrical angle sine (-1..1)
in	<i>cosTheta</i> <sub>e</sub>	Electrical angle cosine (-1..1)
out	<i>idMeas</i>	Pointer to store the D-axis current.
out	<i>iqMeas</i>	Pointer to store the Q-axis current.

Here is the caller graph for this function:



## 4.33.2.4 get\_linear()

```

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

Convert ADC reading to physical measurement with linear response.

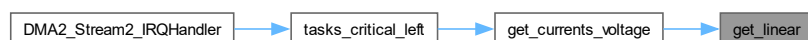
## Parameters

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

## Return values

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

Here is the caller graph for this function:



#### 4.33.2.5 get\_temperature()

```
float get_temperature (
    uint32_t bits,
    const float tempLUT[] )
```

Retrieves temperature from a lookup table based on ADC bits.

This function retrieves temperature from a lookup table based on the ADC bits. The lookup table (LUT) must have a value for each possible ADC bit combination.

##### Parameters

in	<i>bits</i>	ADC reading converted to bits.
in	<i>tempLUT</i>	Lookup table containing temperature values.

##### Returns

Temperature corresponding to the provided ADC bits.

Here is the caller graph for this function:



### 4.33.3 Variable Documentation

#### 4.33.3.1 rawADC\_left

```
volatile uint16_t rawADC_left[4] = {0}
```

Raw ADC data for the left inverter.

External declaration of raw ADC data for the left inverter.

#### 4.33.3.2 rawADC\_right

```
volatile uint16_t rawADC_right[4] = {0}
```

Raw ADC data for the right inverter.

External declaration of raw ADC data for the right inverter.

#### 4.33.3.3 rawADC\_temp

```
volatile uint16_t rawADC_temp[4] = {0}
```

Raw ADC data for the temperatures.

External declaration of raw ADC data for the temperature readings.

#### 4.33.3.4 tempInverterLUT

```
const float tempInverterLUT[] = {-2.45, -2.44, -2.44, -2.43, -2.42, -2.42, -2.41, -2.41, -2.40, -2.39, -2.39, -2.38, -2.37, -2.37, -2.36, -2.36, -2.35, -2.34, -2.34, -2.33, -2.32, -2.31, -2.31, -2.30, -2.29, -2.29, -2.28, -2.27, -2.27, -2.26, -2.26, -2.25, -2.24, -2.24, -2.23, -2.22, -2.22, -2.21, -2.20, -2.20, -2.19, -2.19, -2.18, -2.17, -2.17, -2.16, -2.16, -2.15, -2.15, -2.14, -2.14, -2.13, -2.12, -2.12, -2.11, -2.10, -2.10, -2.09, -2.08, -2.08, -2.07, -2.07, -2.06, -2.05, -2.05, -2.04, -2.03, -2.03, -2.02, -2.01, -2.01, -2.00, -2.00, -1.99, -1.98, -1.98, -1.97, -1.96, -1.96, -1.95, -1.94, -1.94, -1.93, -1.93, -1.92, -1.91, -1.91, -1.90, -1.89, -1.89, -1.88, -1.87, -1.87, -1.86, -1.86, -1.85, -1.84, -1.84, -1.83, -1.83, -1.82, -1.81, -1.80, -1.80, -1.79, -1.78, -1.78, -1.77, -1.77, -1.76, -1.75, -1.75, -1.74, -1.73, -1.73, -1.72, -1.71, -1.71, -1.70, -1.69, -1.69, -1.68, -1.67, -1.67, -1.66, -1.66, -1.65, -1.64, -1.64, -1.63, -1.62, -1.62, -1.61, -1.60, -1.60, -1.59, -1.58, -1.58, -1.57, -1.56, -1.56, -1.55, -1.54, -1.54, -1.53, -1.53, -1.52, -1.51, -1.51, -1.50, -1.49, -1.49, -1.48, -1.47, -1.47, -1.46, -1.45, -1.45, -1.44, -1.43, -1.43, -1.42, -1.41, -1.41, -1.40, -1.39, -1.39, -1.38, -1.37, -1.37, -1.36, -1.36, -1.35, -1.34, -1.34, -1.33, -1.32, -1.32, -1.31, -1.30, -1.30, -1.29, -1.28, -1.28, -1.27, -1.26, -1.26, -1.25, -1.24, -1.24, -1.23, -1.22, -1.22, -1.21, -1.20, -1.20, -1.19, -1.18, -1.18, -1.17, -1.16, -1.16, -1.15, -1.15, -1.14, -1.14, -1.13, -1.12, -1.12, -1.11, -1.10, -1.10, -1.09, -1.08, -1.08, -1.07, -1.06, -1.06, -1.05, -1.04, -1.04, -1.03, -1.02, -1.02, -1.01, -1.00, -1.00, -0.99, -0.98, -0.98, -0.97, -0.96, -0.96, -0.95, -0.94, -0.94, -0.93, -0.92, -0.92, -0.91, -0.90, -0.90, -0.89, -0.89, -0.88, -0.88, -0.87, -0.86, -0.86, -0.85, -0.84, -0.84, -0.83, -0.82, -0.82, -0.81, -0.80, -0.80, -0.79, -0.78, -0.78, -0.77, -0.76, -0.76, -0.75, -0.74, -0.73, -0.73, -0.72, -0.71, -0.71, -0.70, -0.69, -0.69, -0.68, -0.67, -0.67, -0.66, -0.65, -0.65, -0.64, -0.63, -0.63, -0.62, -0.61, -0.61, -0.60, -0.59, -0.59, -0.58, -0.57, -0.56, -0.56, -0.55, -0.54, -0.54, -0.53, -0.52, -0.52, -0.51, -0.50, -0.50, -0.49, -0.48, -0.48, -0.47, -0.46, -0.46, -0.45, -0.45, -0.44, -0.43, -0.43, -0.42, -0.41, -0.41, -0.40, -0.39, -0.39, -0.38, -0.37, -0.37, -0.36, -0.36, -0.35, -0.35, -0.34, -0.33, -0.32, -0.32, -0.31, -0.30, -0.30, -0.29, -0.28, -0.28, -0.27, -0.27, -0.26, -0.26, -0.25, -0.24, -0.24, -0.23, -0.23, -0.22, -0.21, -0.21, -0.20, -0.19, -0.19, -0.18, -0.18, -0.17, -0.17, -0.16, -0.15, -0.15, -0.14, -0.14, -0.13, -0.12, -0.12, -0.11, -0.10, -0.10, -0.09, -0.09, -0.08, -0.08, -0.07, -0.07, -0.06, -0.05, -0.05, -0.04, -0.03, -0.03, -0.02, -0.01, -0.01, -0.00, 0.00, 0.01, 0.01, 0.02, 0.02, 0.03, 0.03, 0.04, 0.04, 0.05, 0.06, 0.07, 0.07, 0.08, 0.09, 0.09, 0.10, 0.11, 0.12, 0.12, 0.13, 0.14, 0.14, 0.15, 0.16, 0.16, 0.17, 0.18, 0.19, 0.19, 0.20, 0.21, 0.21, 0.22, 0.23, 0.24, 0.24, 0.25, 0.26, 0.26, 0.27, 0.28, 0.29, 0.29, 0.30, 0.31, 0.31, 0.32, 0.33, 0.34, 0.34, 0.35, 0.36, 0.36, 0.37, 0.38, 0.39, 0.39, 0.40, 0.41, 0.41, 0.42, 0.43, 0.44, 0.44, 0.45, 0.46, 0.46, 0.47, 0.48, 0.49, 0.49, 0.50, 0.51, 0.51, 0.52, 0.53, 0.54, 0.54, 0.55, 0.56, 0.56, 0.57, 0.58, 0.59, 0.59, 0.60, 0.61, 0.61, 0.62, 0.63, 0.64, 0.64, 0.65, 0.66, 0.67, 0.67, 0.68, 0.69, 0.69, 0.70, 0.71, 0.72, 0.72, 0.73, 0.74, 0.75, 0.75, 0.76, 0.77, 0.77, 0.78, 0.79, 0.80, 0.80, 0.81, 0.82, 0.83, 0.83, 0.84, 0.85, 0.85, 0.86, 0.87, 0.88, 0.88, 0.89, 0.90, 0.91, 0.91, 0.92, 0.93, 0.94, 0.94, 0.95, 0.96, 0.96, 0.97, 0.98, 0.99, 0.99, 1.00, 1.01, 1.02, 1.03, 1.03, 1.04, 1.05, 1.05, 1.06, 1.07, 1.08, 1.08, 1.09, 1.10, 1.10, 1.11, 1.12, 1.13, 1.13, 1.14, 1.15, 1.16, 1.16, 1.17, 1.18, 1.19, 1.19, 1.20, 1.21, 1.22, 1.22, 1.23, 1.24, 1.25, 1.26, 1.26, 1.27, 1.28, 1.28, 1.29, 1.30, 1.31, 1.31, 1.32, 1.33, 1.34, 1.34, 1.35, 1.36, 1.37, 1.37, 1.38, 1.39, 1.40, 1.40, 1.41, 1.42, 1.43, 1.43, 1.44, 1.45, 1.46, 1.46, 1.47, 1.48, 1.49, 1.49, 1.50, 1.51, 1.52, 1.52, 1.53, 1.54, 1.55, 1.55, 1.56, 1.57, 1.58, 1.58, 1.59, 1.60, 1.61, 1.61, 1.62, 1.63, 1.64, 1.64, 1.65, 1.66, 1.67, 1.67, 1.68, 1.69, 1.70, 1.71, 1.71, 1.72, 1.73, 1.74, 1.74, 1.75, 1.76, 1.77, 1.77, 1.78, 1.79, 1.80, 1.80, 1.81, 1.82, 1.83, 1.84,
```

1.84, 1.85, 1.86, 1.87, 1.87, 1.88, 1.89, 1.90, 1.90, 1.91, 1.92, 1.93, 1.93, 1.94, 1.95, 1.↵  
96, 1.97, 1.97, 1.98, 1.99, 2.00, 2.00, 2.01, 2.02, 2.03, 2.04, 2.04, 2.05, 2.06, 2.07, 2.07,  
2.08, 2.09, 2.10, 2.10, 2.11, 2.12, 2.13, 2.14, 2.14, 2.15, 2.16, 2.17, 2.17, 2.18, 2.19, 2.↵  
20, 2.21, 2.21, 2.22, 2.23, 2.24, 2.25, 2.25, 2.26, 2.27, 2.28, 2.28, 2.29, 2.30, 2.31, 2.32,  
2.32, 2.33, 2.34, 2.35, 2.35, 2.36, 2.37, 2.38, 2.39, 2.39, 2.40, 2.41, 2.42, 2.43, 2.43, 2.↵  
44, 2.45, 2.46, 2.46, 2.47, 2.48, 2.49, 2.50, 2.50, 2.51, 2.52, 2.53, 2.54, 2.54, 2.55, 2.56,  
2.57, 2.58, 2.58, 2.59, 2.60, 2.61, 2.62, 2.62, 2.63, 2.64, 2.65, 2.66, 2.66, 2.67, 2.68, 2.↵  
69, 2.70, 2.70, 2.71, 2.72, 2.73, 2.74, 2.74, 2.75, 2.76, 2.77, 2.78, 2.78, 2.79, 2.80, 2.81,  
2.82, 2.82, 2.83, 2.84, 2.85, 2.86, 2.86, 2.87, 2.88, 2.89, 2.90, 2.90, 2.91, 2.92, 2.93, 2.↵  
94, 2.94, 2.95, 2.96, 2.97, 2.98, 2.98, 2.99, 3.00, 3.01, 3.02, 3.02, 3.03, 3.04, 3.05, 3.06,  
3.07, 3.07, 3.08, 3.09, 3.10, 3.11, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.16, 3.17, 3.18, 3.↵  
19, 3.20, 3.20, 3.21, 3.22, 3.23, 3.24, 3.24, 3.25, 3.26, 3.27, 3.28, 3.29, 3.29, 3.30, 3.31,  
3.32, 3.33, 3.34, 3.34, 3.35, 3.36, 3.37, 3.38, 3.38, 3.39, 3.40, 3.41, 3.42, 3.43, 3.43, 3.↵  
44, 3.45, 3.46, 3.47, 3.48, 3.48, 3.49, 3.50, 3.51, 3.52, 3.53, 3.53, 3.54, 3.55, 3.56, 3.57,  
3.58, 3.58, 3.59, 3.60, 3.61, 3.62, 3.63, 3.63, 3.64, 3.65, 3.66, 3.67, 3.68, 3.68, 3.69, 3.↵  
70, 3.71, 3.72, 3.73, 3.73, 3.74, 3.75, 3.76, 3.77, 3.78, 3.78, 3.79, 3.80, 3.81, 3.82, 3.83,  
3.83, 3.84, 3.85, 3.86, 3.87, 3.88, 3.89, 3.89, 3.90, 3.91, 3.92, 3.93, 3.94, 3.94, 3.95, 3.↵  
96, 3.97, 3.98, 3.99, 4.00, 4.00, 4.01, 4.02, 4.03, 4.04, 4.05, 4.05, 4.06, 4.07, 4.08, 4.09,  
4.10, 4.11, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.17, 4.18, 4.19, 4.20, 4.21, 4.22, 4.↵  
23, 4.23, 4.24, 4.25, 4.26, 4.27, 4.28, 4.29, 4.29, 4.30, 4.31, 4.32, 4.33, 4.34, 4.35, 4.35,  
4.36, 4.37, 4.38, 4.39, 4.40, 4.41, 4.42, 4.42, 4.43, 4.44, 4.45, 4.46, 4.47, 4.48, 4.48, 4.↵  
49, 4.50, 4.51, 4.52, 4.53, 4.54, 4.55, 4.55, 4.56, 4.57, 4.58, 4.59, 4.60, 4.61, 4.62, 4.62,  
4.63, 4.64, 4.65, 4.66, 4.67, 4.68, 4.69, 4.69, 4.70, 4.71, 4.72, 4.73, 4.74, 4.75, 4.76, 4.↵  
76, 4.77, 4.78, 4.79, 4.80, 4.81, 4.82, 4.83, 4.83, 4.84, 4.85, 4.86, 4.87, 4.88, 4.89, 4.90,  
4.91, 4.91, 4.92, 4.93, 4.94, 4.95, 4.96, 4.97, 4.98, 4.99, 4.99, 5.00, 5.01, 5.02, 5.03, 5.↵  
04, 5.05, 5.06, 5.07, 5.07, 5.08, 5.09, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.16, 5.17,  
5.18, 5.19, 5.20, 5.21, 5.22, 5.23, 5.24, 5.24, 5.25, 5.26, 5.27, 5.28, 5.29, 5.30, 5.31, 5.↵  
32, 5.33, 5.34, 5.34, 5.35, 5.36, 5.37, 5.38, 5.39, 5.40, 5.41, 5.42, 5.43, 5.43, 5.44, 5.45,  
5.46, 5.47, 5.48, 5.49, 5.50, 5.51, 5.52, 5.53, 5.53, 5.54, 5.55, 5.56, 5.57, 5.58, 5.59, 5.↵  
60, 5.61, 5.62, 5.63, 5.64, 5.64, 5.65, 5.66, 5.67, 5.68, 5.69, 5.70, 5.71, 5.72, 5.73, 5.74,  
5.75, 5.75, 5.76, 5.77, 5.78, 5.79, 5.80, 5.81, 5.82, 5.83, 5.84, 5.85, 5.86, 5.87, 5.88, 5.↵  
88, 5.89, 5.90, 5.91, 5.92, 5.93, 5.94, 5.95, 5.96, 5.97, 5.98, 5.99, 6.00, 6.01, 6.01, 6.02,  
6.03, 6.04, 6.05, 6.06, 6.07, 6.08, 6.09, 6.10, 6.11, 6.12, 6.13, 6.14, 6.15, 6.16, 6.16, 6.↵  
17, 6.18, 6.19, 6.20, 6.21, 6.22, 6.23, 6.24, 6.25, 6.26, 6.27, 6.28, 6.29, 6.30, 6.31, 6.32,  
6.32, 6.33, 6.34, 6.35, 6.36, 6.37, 6.38, 6.39, 6.40, 6.41, 6.42, 6.43, 6.44, 6.45, 6.46, 6.↵  
47, 6.48, 6.49, 6.50, 6.51, 6.51, 6.52, 6.53, 6.54, 6.55, 6.56, 6.57, 6.58, 6.59, 6.60, 6.61,  
6.62, 6.63, 6.64, 6.65, 6.66, 6.67, 6.68, 6.69, 6.70, 6.71, 6.72, 6.73, 6.74, 6.75, 6.75, 6.↵  
76, 6.77, 6.78, 6.79, 6.80, 6.81, 6.82, 6.83, 6.84, 6.85, 6.86, 6.87, 6.88, 6.89, 6.90, 6.91,  
6.92, 6.93, 6.94, 6.95, 6.96, 6.97, 6.98, 6.99, 7.00, 7.01, 7.02, 7.03, 7.04, 7.05, 7.06, 7.↵  
07, 7.08, 7.09, 7.10, 7.11, 7.12, 7.12, 7.13, 7.14, 7.15, 7.16, 7.17, 7.18, 7.19, 7.20, 7.21,  
7.22, 7.23, 7.24, 7.25, 7.26, 7.27, 7.28, 7.29, 7.30, 7.31, 7.32, 7.33, 7.34, 7.35, 7.36, 7.↵  
37, 7.38, 7.39, 7.40, 7.41, 7.42, 7.43, 7.44, 7.45, 7.46, 7.47, 7.48, 7.49, 7.50, 7.51, 7.52,  
7.53, 7.54, 7.55, 7.56, 7.57, 7.58, 7.59, 7.60, 7.61, 7.62, 7.63, 7.64, 7.65, 7.66, 7.67, 7.↵  
68, 7.69, 7.70, 7.71, 7.72, 7.73, 7.74, 7.75, 7.76, 7.77, 7.78, 7.79, 7.80, 7.81, 7.82, 7.83,  
7.84, 7.85, 7.86, 7.87, 7.88, 7.89, 7.91, 7.92, 7.93, 7.94, 7.95, 7.96, 7.97, 7.98, 7.99, 8.↵  
00, 8.01, 8.02, 8.03, 8.04, 8.05, 8.06, 8.07, 8.08, 8.09, 8.10, 8.11, 8.12, 8.13, 8.14, 8.15,  
8.16, 8.17, 8.18, 8.19, 8.20, 8.21, 8.22, 8.23, 8.24, 8.25, 8.26, 8.27, 8.29, 8.30, 8.31, 8.↵  
32, 8.33, 8.34, 8.35, 8.36, 8.37, 8.38, 8.39, 8.40, 8.41, 8.42, 8.43, 8.44, 8.45, 8.46, 8.47,  
8.48, 8.49, 8.50, 8.51, 8.52, 8.54, 8.55, 8.56, 8.57, 8.58, 8.59, 8.60, 8.61, 8.62, 8.63, 8.↵  
64, 8.65, 8.66, 8.67, 8.68, 8.69, 8.70, 8.71, 8.72, 8.74, 8.75, 8.76, 8.77, 8.78, 8.79, 8.80,  
8.81, 8.82, 8.83, 8.84, 8.85, 8.86, 8.87, 8.88, 8.89, 8.91, 8.92, 8.93, 8.94, 8.95, 8.96, 8.↵  
97, 8.98, 8.99, 9.00, 9.01, 9.02, 9.03, 9.04, 9.06, 9.07, 9.08, 9.09, 9.10, 9.11, 9.12, 9.13,  
9.14, 9.15, 9.16, 9.17, 9.18, 9.20, 9.21, 9.22, 9.23, 9.24, 9.25, 9.26, 9.27, 9.28, 9.29, 9.↵  
30, 9.31, 9.33, 9.34, 9.35, 9.36, 9.37, 9.38, 9.39, 9.40, 9.41, 9.42, 9.43, 9.45, 9.46, 9.47,  
9.48, 9.49, 9.50, 9.51, 9.52, 9.53, 9.54, 9.55, 9.57, 9.58, 9.59, 9.60, 9.61, 9.62, 9.63, 9.↵  
64, 9.65, 9.66, 9.68, 9.69, 9.70, 9.71, 9.72, 9.73, 9.74, 9.75, 9.76, 9.78, 9.79, 9.80, 9.81,  
9.82, 9.83, 9.84, 9.85, 9.86, 9.88, 9.89, 9.90, 9.91, 9.92, 9.93, 9.94, 9.95, 9.96, 9.98, 9.↵  
99, 10.00, 10.01, 10.02, 10.03, 10.04, 10.05, 10.07, 10.08, 10.09, 10.10, 10.11, 10.12, 10.↵

13, 10.14, 10.16, 10.17, 10.18, 10.19, 10.20, 10.21, 10.22, 10.24, 10.25, 10.26, 10.27, 10.↵  
28, 10.29, 10.30, 10.31, 10.33, 10.34, 10.35, 10.36, 10.37, 10.38, 10.39, 10.41, 10.42, 10.↵  
43, 10.44, 10.45, 10.46, 10.47, 10.49, 10.50, 10.51, 10.52, 10.53, 10.54, 10.55, 10.57, 10.↵  
58, 10.59, 10.60, 10.61, 10.62, 10.64, 10.65, 10.66, 10.67, 10.68, 10.69, 10.70, 10.72, 10.↵  
73, 10.74, 10.75, 10.76, 10.77, 10.79, 10.80, 10.81, 10.82, 10.83, 10.84, 10.86, 10.87, 10.↵  
88, 10.89, 10.90, 10.91, 10.93, 10.94, 10.95, 10.96, 10.97, 10.98, 11.00, 11.01, 11.02, 11.↵  
03, 11.04, 11.05, 11.07, 11.08, 11.09, 11.10, 11.11, 11.13, 11.14, 11.15, 11.16, 11.17, 11.↵  
18, 11.20, 11.21, 11.22, 11.23, 11.24, 11.26, 11.27, 11.28, 11.29, 11.30, 11.32, 11.33, 11.↵  
34, 11.35, 11.36, 11.37, 11.39, 11.40, 11.41, 11.42, 11.43, 11.45, 11.46, 11.47, 11.48, 11.↵  
49, 11.51, 11.52, 11.53, 11.54, 11.55, 11.57, 11.58, 11.59, 11.60, 11.61, 11.63, 11.64, 11.↵  
65, 11.66, 11.68, 11.69, 11.70, 11.71, 11.72, 11.74, 11.75, 11.76, 11.77, 11.78, 11.80, 11.↵  
81, 11.82, 11.83, 11.85, 11.86, 11.87, 11.88, 11.89, 11.91, 11.92, 11.93, 11.94, 11.96, 11.↵  
97, 11.98, 11.99, 12.00, 12.02, 12.03, 12.04, 12.05, 12.07, 12.08, 12.09, 12.10, 12.11, 12.↵  
13, 12.14, 12.15, 12.16, 12.18, 12.19, 12.20, 12.21, 12.23, 12.24, 12.25, 12.26, 12.28, 12.↵  
29, 12.30, 12.31, 12.33, 12.34, 12.35, 12.36, 12.38, 12.39, 12.40, 12.41, 12.43, 12.44, 12.↵  
45, 12.46, 12.48, 12.49, 12.50, 12.51, 12.53, 12.54, 12.55, 12.56, 12.58, 12.59, 12.60, 12.↵  
61, 12.63, 12.64, 12.65, 12.66, 12.68, 12.69, 12.70, 12.72, 12.73, 12.74, 12.75, 12.77, 12.↵  
78, 12.79, 12.80, 12.82, 12.83, 12.84, 12.86, 12.87, 12.88, 12.89, 12.91, 12.92, 12.93, 12.↵  
94, 12.96, 12.97, 12.98, 13.00, 13.01, 13.02, 13.03, 13.05, 13.06, 13.07, 13.09, 13.10, 13.↵  
11, 13.12, 13.14, 13.15, 13.16, 13.18, 13.19, 13.20, 13.22, 13.23, 13.24, 13.25, 13.27, 13.↵  
28, 13.29, 13.31, 13.32, 13.33, 13.35, 13.36, 13.37, 13.38, 13.40, 13.41, 13.42, 13.44, 13.↵  
45, 13.46, 13.48, 13.49, 13.50, 13.52, 13.53, 13.54, 13.55, 13.57, 13.58, 13.59, 13.61, 13.↵  
62, 13.63, 13.65, 13.66, 13.67, 13.69, 13.70, 13.71, 13.73, 13.74, 13.75, 13.77, 13.78, 13.↵  
79, 13.81, 13.82, 13.83, 13.85, 13.86, 13.87, 13.89, 13.90, 13.91, 13.93, 13.94, 13.95, 13.↵  
97, 13.98, 13.99, 14.01, 14.02, 14.03, 14.05, 14.06, 14.07, 14.09, 14.10, 14.11, 14.13, 14.↵  
14, 14.16, 14.17, 14.18, 14.20, 14.21, 14.22, 14.24, 14.25, 14.26, 14.28, 14.29, 14.30, 14.↵  
32, 14.33, 14.35, 14.36, 14.37, 14.39, 14.40, 14.41, 14.43, 14.44, 14.45, 14.47, 14.48, 14.↵  
50, 14.51, 14.52, 14.54, 14.55, 14.56, 14.58, 14.59, 14.61, 14.62, 14.63, 14.65, 14.66, 14.↵  
67, 14.69, 14.70, 14.72, 14.73, 14.74, 14.76, 14.77, 14.79, 14.80, 14.81, 14.83, 14.84, 14.↵  
86, 14.87, 14.88, 14.90, 14.91, 14.93, 14.94, 14.95, 14.97, 14.98, 15.00, 15.01, 15.02, 15.↵  
04, 15.05, 15.07, 15.08, 15.09, 15.11, 15.12, 15.14, 15.15, 15.16, 15.18, 15.19, 15.21, 15.↵  
22, 15.24, 15.25, 15.26, 15.28, 15.29, 15.31, 15.32, 15.33, 15.35, 15.36, 15.38, 15.39, 15.↵  
41, 15.42, 15.43, 15.45, 15.46, 15.48, 15.49, 15.51, 15.52, 15.54, 15.55, 15.56, 15.58, 15.↵  
59, 15.61, 15.62, 15.64, 15.65, 15.66, 15.68, 15.69, 15.71, 15.72, 15.74, 15.75, 15.77, 15.↵  
78, 15.80, 15.81, 15.82, 15.84, 15.85, 15.87, 15.88, 15.90, 15.91, 15.93, 15.94, 15.96, 15.↵  
97, 15.99, 16.00, 16.01, 16.03, 16.04, 16.06, 16.07, 16.09, 16.10, 16.12, 16.13, 16.15, 16.↵  
16, 16.18, 16.19, 16.21, 16.22, 16.24, 16.25, 16.27, 16.28, 16.30, 16.31, 16.33, 16.34, 16.↵  
35, 16.37, 16.38, 16.40, 16.41, 16.43, 16.44, 16.46, 16.47, 16.49, 16.50, 16.52, 16.53, 16.↵  
55, 16.56, 16.58, 16.59, 16.61, 16.62, 16.64, 16.66, 16.67, 16.69, 16.70, 16.72, 16.73, 16.↵  
75, 16.76, 16.78, 16.79, 16.81, 16.82, 16.84, 16.85, 16.87, 16.88, 16.90, 16.91, 16.93, 16.↵  
94, 16.96, 16.97, 16.99, 17.01, 17.02, 17.04, 17.05, 17.07, 17.08, 17.10, 17.11, 17.13, 17.↵  
14, 17.16, 17.17, 17.19, 17.21, 17.22, 17.24, 17.25, 17.27, 17.28, 17.30, 17.31, 17.33, 17.↵  
35, 17.36, 17.38, 17.39, 17.41, 17.42, 17.44, 17.45, 17.47, 17.49, 17.50, 17.52, 17.53, 17.↵  
55, 17.56, 17.58, 17.60, 17.61, 17.63, 17.64, 17.66, 17.67, 17.69, 17.71, 17.72, 17.74, 17.↵  
75, 17.77, 17.79, 17.80, 17.82, 17.83, 17.85, 17.86, 17.88, 17.90, 17.91, 17.93, 17.94, 17.↵  
96, 17.98, 17.99, 18.01, 18.02, 18.04, 18.06, 18.07, 18.09, 18.11, 18.12, 18.14, 18.15, 18.↵  
17, 18.19, 18.20, 18.22, 18.23, 18.25, 18.27, 18.28, 18.30, 18.32, 18.33, 18.35, 18.36, 18.↵  
38, 18.40, 18.41, 18.43, 18.45, 18.46, 18.48, 18.49, 18.51, 18.53, 18.54, 18.56, 18.58, 18.↵  
59, 18.61, 18.63, 18.64, 18.66, 18.68, 18.69, 18.71, 18.73, 18.74, 18.76, 18.77, 18.79, 18.↵  
81, 18.82, 18.84, 18.86, 18.87, 18.89, 18.91, 18.92, 18.94, 18.96, 18.97, 18.99, 19.01, 19.↵  
02, 19.04, 19.06, 19.08, 19.09, 19.11, 19.13, 19.14, 19.16, 19.18, 19.19, 19.21, 19.23, 19.↵  
24, 19.26, 19.28, 19.29, 19.31, 19.33, 19.35, 19.36, 19.38, 19.40, 19.41, 19.43, 19.45, 19.↵  
46, 19.48, 19.50, 19.52, 19.53, 19.55, 19.57, 19.58, 19.60, 19.62, 19.64, 19.65, 19.67, 19.↵  
69, 19.70, 19.72, 19.74, 19.76, 19.77, 19.79, 19.81, 19.83, 19.84, 19.86, 19.88, 19.90, 19.↵  
91, 19.93, 19.95, 19.97, 19.98, 20.00, 20.02, 20.04, 20.05, 20.07, 20.09, 20.11, 20.12, 20.↵  
14, 20.16, 20.18, 20.19, 20.21, 20.23, 20.25, 20.26, 20.28, 20.30, 20.32, 20.33, 20.35, 20.↵  
37, 20.39, 20.41, 20.42, 20.44, 20.46, 20.48, 20.49, 20.51, 20.53, 20.55, 20.57, 20.58, 20.↵  
60, 20.62, 20.64, 20.66, 20.67, 20.69, 20.71, 20.73, 20.75, 20.76, 20.78, 20.80, 20.82, 20.↵

84, 20.85, 20.87, 20.89, 20.91, 20.93, 20.95, 20.96, 20.98, 21.00, 21.02, 21.04, 21.06, 21.↵  
07, 21.09, 21.11, 21.13, 21.15, 21.17, 21.18, 21.20, 21.22, 21.24, 21.26, 21.28, 21.29, 21.↵  
31, 21.33, 21.35, 21.37, 21.39, 21.41, 21.42, 21.44, 21.46, 21.48, 21.50, 21.52, 21.54, 21.↵  
55, 21.57, 21.59, 21.61, 21.63, 21.65, 21.67, 21.69, 21.70, 21.72, 21.74, 21.76, 21.78, 21.↵  
80, 21.82, 21.84, 21.86, 21.87, 21.89, 21.91, 21.93, 21.95, 21.97, 21.99, 22.01, 22.03, 22.↵  
05, 22.06, 22.08, 22.10, 22.12, 22.14, 22.16, 22.18, 22.20, 22.22, 22.24, 22.26, 22.28, 22.↵  
30, 22.31, 22.33, 22.35, 22.37, 22.39, 22.41, 22.43, 22.45, 22.47, 22.49, 22.51, 22.53, 22.↵  
55, 22.57, 22.59, 22.61, 22.63, 22.64, 22.66, 22.68, 22.70, 22.72, 22.74, 22.76, 22.78, 22.↵  
80, 22.82, 22.84, 22.86, 22.88, 22.90, 22.92, 22.94, 22.96, 22.98, 23.00, 23.02, 23.04, 23.↵  
06, 23.08, 23.10, 23.12, 23.14, 23.16, 23.18, 23.20, 23.22, 23.24, 23.26, 23.28, 23.30, 23.↵  
32, 23.34, 23.36, 23.38, 23.40, 23.42, 23.44, 23.46, 23.48, 23.50, 23.52, 23.54, 23.56, 23.↵  
58, 23.60, 23.62, 23.65, 23.67, 23.69, 23.71, 23.73, 23.75, 23.77, 23.79, 23.81, 23.83, 23.↵  
85, 23.87, 23.89, 23.91, 23.93, 23.95, 23.97, 24.00, 24.02, 24.04, 24.06, 24.08, 24.10, 24.↵  
12, 24.14, 24.16, 24.18, 24.20, 24.22, 24.25, 24.27, 24.29, 24.31, 24.33, 24.35, 24.37, 24.↵  
39, 24.41, 24.43, 24.46, 24.48, 24.50, 24.52, 24.54, 24.56, 24.58, 24.60, 24.63, 24.65, 24.↵  
67, 24.69, 24.71, 24.73, 24.75, 24.78, 24.80, 24.82, 24.84, 24.86, 24.88, 24.90, 24.93, 24.↵  
95, 24.97, 24.99, 25.01, 25.03, 25.06, 25.08, 25.10, 25.12, 25.14, 25.16, 25.19, 25.21, 25.↵  
23, 25.25, 25.27, 25.30, 25.32, 25.34, 25.36, 25.38, 25.41, 25.43, 25.45, 25.47, 25.49, 25.↵  
52, 25.54, 25.56, 25.58, 25.60, 25.63, 25.65, 25.67, 25.69, 25.72, 25.74, 25.76, 25.78, 25.↵  
81, 25.83, 25.85, 25.87, 25.89, 25.92, 25.94, 25.96, 25.98, 26.01, 26.03, 26.05, 26.08, 26.↵  
10, 26.12, 26.14, 26.17, 26.19, 26.21, 26.23, 26.26, 26.28, 26.30, 26.33, 26.35, 26.37, 26.↵  
39, 26.42, 26.44, 26.46, 26.49, 26.51, 26.53, 26.56, 26.58, 26.60, 26.63, 26.65, 26.67, 26.↵  
69, 26.72, 26.74, 26.76, 26.79, 26.81, 26.83, 26.86, 26.88, 26.90, 26.93, 26.95, 26.98, 27.↵  
00, 27.02, 27.05, 27.07, 27.09, 27.12, 27.14, 27.16, 27.19, 27.21, 27.24, 27.26, 27.28, 27.↵  
31, 27.33, 27.35, 27.38, 27.40, 27.43, 27.45, 27.47, 27.50, 27.52, 27.55, 27.57, 27.59, 27.↵  
62, 27.64, 27.67, 27.69, 27.72, 27.74, 27.76, 27.79, 27.81, 27.84, 27.86, 27.89, 27.91, 27.↵  
93, 27.96, 27.98, 28.01, 28.03, 28.06, 28.08, 28.11, 28.13, 28.16, 28.18, 28.21, 28.23, 28.↵  
26, 28.28, 28.30, 28.33, 28.35, 28.38, 28.40, 28.43, 28.45, 28.48, 28.50, 28.53, 28.55, 28.↵  
58, 28.60, 28.63, 28.66, 28.68, 28.71, 28.73, 28.76, 28.78, 28.81, 28.83, 28.86, 28.88, 28.↵  
91, 28.93, 28.96, 28.99, 29.01, 29.04, 29.06, 29.09, 29.11, 29.14, 29.17, 29.19, 29.22, 29.↵  
24, 29.27, 29.29, 29.32, 29.35, 29.37, 29.40, 29.42, 29.45, 29.48, 29.50, 29.53, 29.55, 29.↵  
58, 29.61, 29.63, 29.66, 29.69, 29.71, 29.74, 29.76, 29.79, 29.82, 29.84, 29.87, 29.90, 29.↵  
92, 29.95, 29.98, 30.00, 30.03, 30.06, 30.08, 30.11, 30.14, 30.16, 30.19, 30.22, 30.24, 30.↵  
27, 30.30, 30.33, 30.35, 30.38, 30.41, 30.43, 30.46, 30.49, 30.52, 30.54, 30.57, 30.60, 30.↵  
62, 30.65, 30.68, 30.71, 30.73, 30.76, 30.79, 30.82, 30.84, 30.87, 30.90, 30.93, 30.96, 30.↵  
98, 31.01, 31.04, 31.07, 31.09, 31.12, 31.15, 31.18, 31.21, 31.23, 31.26, 31.29, 31.32, 31.↵  
35, 31.37, 31.40, 31.43, 31.46, 31.49, 31.52, 31.54, 31.57, 31.60, 31.63, 31.66, 31.69, 31.↵  
72, 31.74, 31.77, 31.80, 31.83, 31.86, 31.89, 31.92, 31.95, 31.97, 32.00, 32.03, 32.06, 32.↵  
09, 32.12, 32.15, 32.18, 32.21, 32.24, 32.27, 32.29, 32.32, 32.35, 32.38, 32.41, 32.44, 32.↵  
47, 32.50, 32.53, 32.56, 32.59, 32.62, 32.65, 32.68, 32.71, 32.74, 32.77, 32.80, 32.83, 32.↵  
86, 32.89, 32.92, 32.95, 32.98, 33.01, 33.04, 33.07, 33.10, 33.13, 33.16, 33.19, 33.22, 33.↵  
25, 33.28, 33.31, 33.34, 33.37, 33.40, 33.43, 33.46, 33.49, 33.53, 33.56, 33.59, 33.62, 33.↵  
65, 33.68, 33.71, 33.74, 33.77, 33.80, 33.84, 33.87, 33.90, 33.93, 33.96, 33.99, 34.02, 34.↵  
05, 34.09, 34.12, 34.15, 34.18, 34.21, 34.24, 34.28, 34.31, 34.34, 34.37, 34.40, 34.43, 34.↵  
47, 34.50, 34.53, 34.56, 34.59, 34.63, 34.66, 34.69, 34.72, 34.76, 34.79, 34.82, 34.85, 34.↵  
89, 34.92, 34.95, 34.98, 35.02, 35.05, 35.08, 35.11, 35.15, 35.18, 35.21, 35.25, 35.28, 35.↵  
31, 35.35, 35.38, 35.41, 35.44, 35.48, 35.51, 35.54, 35.58, 35.61, 35.65, 35.68, 35.71, 35.↵  
75, 35.78, 35.81, 35.85, 35.88, 35.91, 35.95, 35.98, 36.02, 36.05, 36.08, 36.12, 36.15, 36.↵  
19, 36.22, 36.26, 36.29, 36.33, 36.36, 36.39, 36.43, 36.46, 36.50, 36.53, 36.57, 36.60, 36.↵  
64, 36.67, 36.71, 36.74, 36.78, 36.81, 36.85, 36.88, 36.92, 36.95, 36.99, 37.02, 37.06, 37.↵  
09, 37.13, 37.17, 37.20, 37.24, 37.27, 37.31, 37.34, 37.38, 37.42, 37.45, 37.49, 37.52, 37.↵  
56, 37.60, 37.63, 37.67, 37.71, 37.74, 37.78, 37.82, 37.85, 37.89, 37.93, 37.96, 38.00, 38.↵  
04, 38.07, 38.11, 38.15, 38.18, 38.22, 38.26, 38.30, 38.33, 38.37, 38.41, 38.44, 38.48, 38.↵  
52, 38.56, 38.60, 38.63, 38.67, 38.71, 38.75, 38.78, 38.82, 38.86, 38.90, 38.94, 38.97, 39.↵  
01, 39.05, 39.09, 39.13, 39.17, 39.21, 39.24, 39.28, 39.32, 39.36, 39.40, 39.44, 39.48, 39.↵  
52, 39.56, 39.59, 39.63, 39.67, 39.71, 39.75, 39.79, 39.83, 39.87, 39.91, 39.95, 39.99, 40.↵  
03, 40.07, 40.11, 40.15, 40.19, 40.23, 40.27, 40.31, 40.35, 40.39, 40.43, 40.47, 40.51, 40.↵  
55, 40.59, 40.64, 40.68, 40.72, 40.76, 40.80, 40.84, 40.88, 40.92, 40.96, 41.01, 41.05, 41.↵



09, 41.13, 41.17, 41.21, 41.26, 41.30, 41.34, 41.38, 41.42, 41.47, 41.51, 41.55, 41.59, 41.↵  
64, 41.68, 41.72, 41.76, 41.81, 41.85, 41.89, 41.93, 41.98, 42.02, 42.06, 42.11, 42.15, 42.↵  
19, 42.24, 42.28, 42.32, 42.37, 42.41, 42.46, 42.50, 42.54, 42.59, 42.63, 42.68, 42.72, 42.↵  
76, 42.81, 42.85, 42.90, 42.94, 42.99, 43.03, 43.08, 43.12, 43.17, 43.21, 43.26, 43.30, 43.↵  
35, 43.39, 43.44, 43.48, 43.53, 43.58, 43.62, 43.67, 43.71, 43.76, 43.81, 43.85, 43.90, 43.↵  
94, 43.99, 44.04, 44.08, 44.13, 44.18, 44.23, 44.27, 44.32, 44.37, 44.41, 44.46, 44.51, 44.↵  
56, 44.60, 44.65, 44.70, 44.75, 44.80, 44.84, 44.89, 44.94, 44.99, 45.04, 45.08, 45.13, 45.↵  
18, 45.23, 45.28, 45.33, 45.38, 45.43, 45.48, 45.53, 45.57, 45.62, 45.67, 45.72, 45.77, 45.↵  
82, 45.87, 45.92, 45.97, 46.02, 46.07, 46.12, 46.17, 46.23, 46.28, 46.33, 46.38, 46.43, 46.↵  
48, 46.53, 46.58, 46.63, 46.69, 46.74, 46.79, 46.84, 46.89, 46.95, 47.00, 47.05, 47.10, 47.↵  
15, 47.21, 47.26, 47.31, 47.37, 47.42, 47.47, 47.52, 47.58, 47.63, 47.69, 47.74, 47.79, 47.↵  
85, 47.90, 47.95, 48.01, 48.06, 48.12, 48.17, 48.23, 48.28, 48.34, 48.39, 48.45, 48.50, 48.↵  
56, 48.61, 48.67, 48.72, 48.78, 48.84, 48.89, 48.95, 49.00, 49.06, 49.12, 49.17, 49.23, 49.↵  
29, 49.34, 49.40, 49.46, 49.52, 49.57, 49.63, 49.69, 49.75, 49.80, 49.86, 49.92, 49.98, 50.↵  
04, 50.10, 50.16, 50.21, 50.27, 50.33, 50.39, 50.45, 50.51, 50.57, 50.63, 50.69, 50.75, 50.↵  
81, 50.87, 50.93, 50.99, 51.05, 51.11, 51.17, 51.24, 51.30, 51.36, 51.42, 51.48, 51.54, 51.↵  
61, 51.67, 51.73, 51.79, 51.86, 51.92, 51.98, 52.04, 52.11, 52.17, 52.23, 52.30, 52.36, 52.↵  
43, 52.49, 52.55, 52.62, 52.68, 52.75, 52.81, 52.88, 52.94, 53.01, 53.07, 53.14, 53.21, 53.↵  
27, 53.34, 53.40, 53.47, 53.54, 53.60, 53.67, 53.74, 53.80, 53.87, 53.94, 54.01, 54.08, 54.↵  
14, 54.21, 54.28, 54.35, 54.42, 54.49, 54.56, 54.63, 54.70, 54.76, 54.83, 54.90, 54.98, 55.↵  
05, 55.12, 55.19, 55.26, 55.33, 55.40, 55.47, 55.54, 55.62, 55.69, 55.76, 55.83, 55.91, 55.↵  
98, 56.05, 56.12, 56.20, 56.27, 56.35, 56.42, 56.49, 56.57, 56.64, 56.72, 56.79, 56.87, 56.↵  
94, 57.02, 57.10, 57.17, 57.25, 57.32, 57.40, 57.48, 57.56, 57.63, 57.71, 57.79, 57.87, 57.↵  
94, 58.02, 58.10, 58.18, 58.26, 58.34, 58.42, 58.50, 58.58, 58.66, 58.74, 58.82, 58.90, 58.↵  
98, 59.06, 59.15, 59.23, 59.31, 59.39, 59.48, 59.56, 59.64, 59.72, 59.81, 59.89, 59.98, 60.↵  
06, 60.15, 60.23, 60.32, 60.40, 60.49, 60.57, 60.66, 60.75, 60.83, 60.92, 61.01, 61.10, 61.↵  
18, 61.27, 61.36, 61.45, 61.54, 61.63, 61.72, 61.81, 61.90, 61.99, 62.08, 62.17, 62.26, 62.↵  
35, 62.44, 62.54, 62.63, 62.72, 62.82, 62.91, 63.00, 63.10, 63.19, 63.29, 63.38, 63.48, 63.↵  
57, 63.67, 63.76, 63.86, 63.96, 64.06, 64.15, 64.25, 64.35, 64.45, 64.55, 64.65, 64.75, 64.↵  
85, 64.95, 65.05, 65.15, 65.25, 65.35, 65.46, 65.56, 65.66, 65.76, 65.87, 65.97, 66.08, 66.↵  
18, 66.29, 66.39, 66.50, 66.61, 66.71, 66.82, 66.93, 67.03, 67.14, 67.25, 67.36, 67.47, 67.↵  
58, 67.69, 67.80, 67.91, 68.03, 68.14, 68.25, 68.36, 68.48, 68.59, 68.71, 68.82, 68.94, 69.↵  
05, 69.17, 69.29, 69.40, 69.52, 69.64, 69.76, 69.88, 70.00, 70.12, 70.24, 70.36, 70.48, 70.↵  
60, 70.73, 70.85, 70.97, 71.10, 71.22, 71.35, 71.47, 71.60, 71.73, 71.86, 71.98, 72.11, 72.↵  
24, 72.37, 72.50, 72.63, 72.76, 72.90, 73.03, 73.16, 73.30, 73.43, 73.57, 73.70, 73.84, 73.↵  
98, 74.11, 74.25, 74.39, 74.53, 74.67, 74.81, 74.95, 75.10, 75.24, 75.38, 75.53, 75.67, 75.↵  
82, 75.97, 76.11, 76.26, 76.41, 76.56, 76.71, 76.86, 77.01, 77.17, 77.32, 77.47, 77.63, 77.↵  
78, 77.94, 78.10, 78.26, 78.42, 78.58, 78.74, 78.90, 79.06, 79.23, 79.39, 79.55, 79.72, 79.↵  
89, 80.06, 80.23, 80.40, 80.57, 80.74, 80.91, 81.08, 81.26, 81.44, 81.61, 81.79, 81.97, 82.↵  
15, 82.33, 82.51, 82.70, 82.88, 83.07, 83.25, 83.44, 83.63, 83.82, 84.01, 84.20, 84.40, 84.↵  
59, 84.79, 84.98, 85.18, 85.38, 85.58, 85.79, 85.99, 86.20, 86.40, 86.61, 86.82, 87.03, 87.↵  
24, 87.46, 87.67, 87.89, 88.11, 88.33, 88.55, 88.77, 89.00, 89.22, 89.45, 89.68, 89.91, 90.↵  
14, 90.38, 90.62, 90.85, 91.09, 91.34, 91.58, 91.82, 92.07, 92.32, 92.57, 92.83, 93.08, 93.↵  
34, 93.60, 93.86, 94.13, 94.39, 94.66, 94.93, 95.20, 95.48, 95.76, 96.04, 96.32, 96.61, 96.89,  
97.18, 97.48, 97.77, 98.07, 98.37, 98.68, 98.98, 99.29, 99.61, 99.92, 100.24, 100.56, 100.89,  
101.22, 101.55, 101.88, 102.22, 102.56, 102.91, 103.26, 103.61, 103.97, 104.33, 104.70, 105.↵  
07, 105.44, 105.82, 106.20, 106.58, 106.98, 107.37, 107.77, 108.18, 108.59, 109.00, 109.42,  
109.85, 110.28, 110.71, 111.16, 111.60, 112.06, 112.52, 112.99, 113.46, 113.94, 114.43, 114.↵  
92, 115.42, 115.93, 116.44, 116.97, 117.50, 118.04, 118.59, 119.14, 119.71, 120.29, 120.87,  
121.47, 122.07, 122.69, 123.32, 123.95, 124.61, 125.27, 125.94, 126.63, 127.33, 128.05, 128.↵  
78, 129.52, 130.29, 131.06, 131.86, 132.67, 133.50, 134.35, 135.22, 136.12, 137.03, 137.97,  
138.93, 139.92, 140.93, 141.97, 143.05, 144.15, 145.29, 146.46, 147.67, 148.92, 150.22, 151.↵  
55, 152.94, 154.38, 155.87, 157.42, 159.04, 160.72, 162.48, 164.33, 166.26, 168.29, 170.42,  
172.68, 175.07, 177.60, 180.30, 183.19, 186.29, 189.63, 193.26, 197.23, 201.59, 206.43, 211.↵  
86, 218.04, 225.19, 233.64, 243.93, 257.02, 274.82, 302.24, 360.00, 421.98, 421.98, 421.98,  
421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98,  
98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98,  
421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98, 421.98,

#### 4.33.3.5 tempMotorLUT

Generated by Doxygen

44, -0.43, -0.43, -0.42, -0.41, -0.41, -0.40, -0.39, -0.39, -0.38, -0.37, -0.37, -0.36, -0.↵  
35, -0.35, -0.34, -0.33, -0.32, -0.32, -0.31, -0.30, -0.30, -0.29, -0.28, -0.28, -0.27, -0.↵  
26, -0.26, -0.25, -0.24, -0.23, -0.23, -0.22, -0.21, -0.21, -0.20, -0.19, -0.19, -0.18, -0.↵  
17, -0.17, -0.16, -0.15, -0.14, -0.14, -0.13, -0.12, -0.12, -0.11, -0.10, -0.10, -0.09, -0.08,  
-0.07, -0.07, -0.06, -0.05, -0.05, -0.04, -0.03, -0.03, -0.02, -0.01, -0.00, 0.00, 0.01, 0.02,  
0.02, 0.03, 0.04, 0.04, 0.05, 0.06, 0.07, 0.07, 0.08, 0.09, 0.09, 0.10, 0.11, 0.12, 0.12, 0.↵  
13, 0.14, 0.14, 0.15, 0.16, 0.16, 0.17, 0.18, 0.19, 0.19, 0.20, 0.21, 0.21, 0.22, 0.23, 0.24,  
0.24, 0.25, 0.26, 0.26, 0.27, 0.28, 0.29, 0.29, 0.30, 0.31, 0.31, 0.32, 0.33, 0.34, 0.34, 0.↵  
35, 0.36, 0.36, 0.37, 0.38, 0.39, 0.39, 0.40, 0.41, 0.41, 0.42, 0.43, 0.44, 0.44, 0.45, 0.46,  
0.46, 0.47, 0.48, 0.49, 0.49, 0.50, 0.51, 0.51, 0.52, 0.53, 0.54, 0.54, 0.55, 0.56, 0.56, 0.↵  
57, 0.58, 0.59, 0.59, 0.60, 0.61, 0.61, 0.62, 0.63, 0.64, 0.64, 0.65, 0.66, 0.67, 0.67, 0.68,  
0.69, 0.69, 0.70, 0.71, 0.72, 0.72, 0.73, 0.74, 0.75, 0.75, 0.76, 0.77, 0.77, 0.78, 0.79, 0.↵  
80, 0.80, 0.81, 0.82, 0.83, 0.83, 0.84, 0.85, 0.85, 0.86, 0.87, 0.88, 0.88, 0.89, 0.90, 0.91,  
0.91, 0.92, 0.93, 0.94, 0.94, 0.95, 0.96, 0.96, 0.97, 0.98, 0.99, 0.99, 1.00, 1.01, 1.02, 1.↵  
02, 1.03, 1.04, 1.05, 1.05, 1.06, 1.07, 1.08, 1.08, 1.09, 1.10, 1.10, 1.11, 1.12, 1.13, 1.13,  
1.14, 1.15, 1.16, 1.16, 1.17, 1.18, 1.19, 1.19, 1.20, 1.21, 1.22, 1.22, 1.23, 1.24, 1.25, 1.↵  
25, 1.26, 1.27, 1.28, 1.28, 1.29, 1.30, 1.31, 1.31, 1.32, 1.33, 1.34, 1.34, 1.35, 1.36, 1.37,  
1.37, 1.38, 1.39, 1.40, 1.40, 1.41, 1.42, 1.43, 1.43, 1.44, 1.45, 1.46, 1.46, 1.47, 1.48, 1.↵  
49, 1.49, 1.50, 1.51, 1.52, 1.52, 1.53, 1.54, 1.55, 1.55, 1.56, 1.57, 1.58, 1.58, 1.59, 1.60,  
1.61, 1.61, 1.62, 1.63, 1.64, 1.64, 1.65, 1.66, 1.67, 1.67, 1.68, 1.69, 1.70, 1.71, 1.71, 1.↵  
72, 1.73, 1.74, 1.74, 1.75, 1.76, 1.77, 1.77, 1.78, 1.79, 1.80, 1.80, 1.81, 1.82, 1.83, 1.84,  
1.84, 1.85, 1.86, 1.87, 1.87, 1.88, 1.89, 1.90, 1.90, 1.91, 1.92, 1.93, 1.93, 1.94, 1.95, 1.↵  
96, 1.97, 1.97, 1.98, 1.99, 2.00, 2.00, 2.01, 2.02, 2.03, 2.04, 2.04, 2.05, 2.06, 2.07, 2.07,  
2.08, 2.09, 2.10, 2.10, 2.11, 2.12, 2.13, 2.14, 2.14, 2.15, 2.16, 2.17, 2.17, 2.18, 2.19, 2.↵  
20, 2.21, 2.21, 2.22, 2.23, 2.24, 2.25, 2.25, 2.26, 2.27, 2.28, 2.28, 2.29, 2.30, 2.31, 2.32,  
2.32, 2.33, 2.34, 2.35, 2.35, 2.36, 2.37, 2.38, 2.39, 2.39, 2.40, 2.41, 2.42, 2.43, 2.43, 2.↵  
44, 2.45, 2.46, 2.46, 2.47, 2.48, 2.49, 2.50, 2.50, 2.51, 2.52, 2.53, 2.54, 2.54, 2.55, 2.56,  
2.57, 2.58, 2.58, 2.59, 2.60, 2.61, 2.62, 2.62, 2.63, 2.64, 2.65, 2.66, 2.66, 2.67, 2.68, 2.↵  
69, 2.70, 2.70, 2.71, 2.72, 2.73, 2.74, 2.74, 2.75, 2.76, 2.77, 2.78, 2.78, 2.79, 2.80, 2.81,  
2.82, 2.82, 2.83, 2.84, 2.85, 2.86, 2.86, 2.87, 2.88, 2.89, 2.90, 2.90, 2.91, 2.92, 2.93, 2.↵  
94, 2.94, 2.95, 2.96, 2.97, 2.98, 2.98, 2.99, 3.00, 3.01, 3.02, 3.02, 3.03, 3.04, 3.05, 3.06,  
3.07, 3.07, 3.08, 3.09, 3.10, 3.11, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.16, 3.17, 3.18, 3.↵  
19, 3.20, 3.20, 3.21, 3.22, 3.23, 3.24, 3.24, 3.25, 3.26, 3.27, 3.28, 3.29, 3.29, 3.30, 3.31,  
3.32, 3.33, 3.34, 3.34, 3.35, 3.36, 3.37, 3.38, 3.38, 3.39, 3.40, 3.41, 3.42, 3.43, 3.43, 3.↵  
44, 3.45, 3.46, 3.47, 3.48, 3.48, 3.49, 3.50, 3.51, 3.52, 3.53, 3.53, 3.54, 3.55, 3.56, 3.57,  
3.58, 3.58, 3.59, 3.60, 3.61, 3.62, 3.63, 3.63, 3.64, 3.65, 3.66, 3.67, 3.68, 3.68, 3.69, 3.↵  
70, 3.71, 3.72, 3.73, 3.73, 3.74, 3.75, 3.76, 3.77, 3.78, 3.78, 3.79, 3.80, 3.81, 3.82, 3.83,  
3.83, 3.84, 3.85, 3.86, 3.87, 3.88, 3.89, 3.89, 3.90, 3.91, 3.92, 3.93, 3.94, 3.94, 3.95, 3.↵  
96, 3.97, 3.98, 3.99, 4.00, 4.00, 4.01, 4.02, 4.03, 4.04, 4.05, 4.05, 4.06, 4.07, 4.08, 4.09,  
4.10, 4.11, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.17, 4.18, 4.19, 4.20, 4.21, 4.22, 4.↵  
23, 4.23, 4.24, 4.25, 4.26, 4.27, 4.28, 4.29, 4.29, 4.30, 4.31, 4.32, 4.33, 4.34, 4.35, 4.35,  
4.36, 4.37, 4.38, 4.39, 4.40, 4.41, 4.42, 4.42, 4.43, 4.44, 4.45, 4.46, 4.47, 4.48, 4.48, 4.↵  
49, 4.50, 4.51, 4.52, 4.53, 4.54, 4.55, 4.55, 4.56, 4.57, 4.58, 4.59, 4.60, 4.61, 4.62, 4.62,  
4.63, 4.64, 4.65, 4.66, 4.67, 4.68, 4.69, 4.69, 4.70, 4.71, 4.72, 4.73, 4.74, 4.75, 4.76, 4.↵  
76, 4.77, 4.78, 4.79, 4.80, 4.81, 4.82, 4.83, 4.83, 4.84, 4.85, 4.86, 4.87, 4.88, 4.89, 4.90,  
4.91, 4.91, 4.92, 4.93, 4.94, 4.95, 4.96, 4.97, 4.98, 4.99, 4.99, 5.00, 5.01, 5.02, 5.03, 5.↵  
04, 5.05, 5.06, 5.07, 5.07, 5.08, 5.09, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.16, 5.17,  
5.18, 5.19, 5.20, 5.21, 5.22, 5.23, 5.24, 5.24, 5.25, 5.26, 5.27, 5.28, 5.29, 5.30, 5.31, 5.↵  
32, 5.33, 5.34, 5.34, 5.35, 5.36, 5.37, 5.38, 5.39, 5.40, 5.41, 5.42, 5.43, 5.43, 5.44, 5.45,  
5.46, 5.47, 5.48, 5.49, 5.50, 5.51, 5.52, 5.53, 5.53, 5.54, 5.55, 5.56, 5.57, 5.58, 5.59, 5.↵  
60, 5.61, 5.62, 5.63, 5.64, 5.64, 5.65, 5.66, 5.67, 5.68, 5.69, 5.70, 5.71, 5.72, 5.73, 5.74,  
5.75, 5.75, 5.76, 5.77, 5.78, 5.79, 5.80, 5.81, 5.82, 5.83, 5.84, 5.85, 5.86, 5.87, 5.88, 5.↵  
88, 5.89, 5.90, 5.91, 5.92, 5.93, 5.94, 5.95, 5.96, 5.97, 5.98, 5.99, 6.00, 6.01, 6.01, 6.02,  
6.03, 6.04, 6.05, 6.06, 6.07, 6.08, 6.09, 6.10, 6.11, 6.12, 6.13, 6.14, 6.15, 6.16, 6.16, 6.↵  
17, 6.18, 6.19, 6.20, 6.21, 6.22, 6.23, 6.24, 6.25, 6.26, 6.27, 6.28, 6.29, 6.30, 6.31, 6.32,  
6.32, 6.33, 6.34, 6.35, 6.36, 6.37, 6.38, 6.39, 6.40, 6.41, 6.42, 6.43, 6.44, 6.45, 6.46, 6.↵  
47, 6.48, 6.49, 6.50, 6.51, 6.51, 6.52, 6.53, 6.54, 6.55, 6.56, 6.57, 6.58, 6.59, 6.60, 6.61,  
6.62, 6.63, 6.64, 6.65, 6.66, 6.67, 6.68, 6.69, 6.70, 6.71, 6.72, 6.73, 6.74, 6.75, 6.↵

76, 6.77, 6.78, 6.79, 6.80, 6.81, 6.82, 6.83, 6.84, 6.85, 6.86, 6.87, 6.88, 6.89, 6.90, 6.91, 6.92, 6.93, 6.94, 6.95, 6.96, 6.97, 6.98, 6.99, 7.00, 7.01, 7.02, 7.03, 7.04, 7.05, 7.06, 7.07, 7.08, 7.09, 7.10, 7.11, 7.12, 7.12, 7.13, 7.14, 7.15, 7.16, 7.17, 7.18, 7.19, 7.20, 7.21, 7.22, 7.23, 7.24, 7.25, 7.26, 7.27, 7.28, 7.29, 7.30, 7.31, 7.32, 7.33, 7.34, 7.35, 7.36, 7.37, 7.38, 7.39, 7.40, 7.41, 7.42, 7.43, 7.44, 7.45, 7.46, 7.47, 7.48, 7.49, 7.50, 7.51, 7.52, 7.53, 7.54, 7.55, 7.56, 7.57, 7.58, 7.59, 7.60, 7.61, 7.62, 7.63, 7.64, 7.65, 7.66, 7.67, 7.68, 7.69, 7.70, 7.71, 7.72, 7.73, 7.74, 7.75, 7.76, 7.77, 7.78, 7.79, 7.80, 7.81, 7.82, 7.83, 7.84, 7.85, 7.86, 7.87, 7.88, 7.89, 7.91, 7.92, 7.93, 7.94, 7.95, 7.96, 7.97, 7.98, 7.99, 8.00, 8.01, 8.02, 8.03, 8.04, 8.05, 8.06, 8.07, 8.08, 8.09, 8.10, 8.11, 8.12, 8.13, 8.14, 8.15, 8.16, 8.17, 8.18, 8.19, 8.20, 8.21, 8.22, 8.23, 8.24, 8.25, 8.26, 8.27, 8.29, 8.30, 8.31, 8.32, 8.33, 8.34, 8.35, 8.36, 8.37, 8.38, 8.39, 8.40, 8.41, 8.42, 8.43, 8.44, 8.45, 8.46, 8.47, 8.48, 8.49, 8.50, 8.51, 8.52, 8.54, 8.55, 8.56, 8.57, 8.58, 8.59, 8.60, 8.61, 8.62, 8.63, 8.64, 8.65, 8.66, 8.67, 8.68, 8.69, 8.70, 8.71, 8.72, 8.74, 8.75, 8.76, 8.77, 8.78, 8.79, 8.80, 8.81, 8.82, 8.83, 8.84, 8.85, 8.86, 8.87, 8.88, 8.89, 8.91, 8.92, 8.93, 8.94, 8.95, 8.96, 8.97, 8.98, 8.99, 9.00, 9.01, 9.02, 9.03, 9.04, 9.06, 9.07, 9.08, 9.09, 9.10, 9.11, 9.12, 9.13, 9.14, 9.15, 9.16, 9.17, 9.18, 9.20, 9.21, 9.22, 9.23, 9.24, 9.25, 9.26, 9.27, 9.28, 9.29, 9.30, 9.31, 9.33, 9.34, 9.35, 9.36, 9.37, 9.38, 9.39, 9.40, 9.41, 9.42, 9.43, 9.45, 9.46, 9.47, 9.48, 9.49, 9.50, 9.51, 9.52, 9.53, 9.54, 9.55, 9.57, 9.58, 9.59, 9.60, 9.61, 9.62, 9.63, 9.64, 9.65, 9.66, 9.68, 9.69, 9.70, 9.71, 9.72, 9.73, 9.74, 9.75, 9.76, 9.78, 9.79, 9.80, 9.81, 9.82, 9.83, 9.84, 9.85, 9.86, 9.88, 9.89, 9.90, 9.91, 9.92, 9.93, 9.94, 9.95, 9.96, 9.98, 9.99, 10.00, 10.01, 10.02, 10.03, 10.04, 10.05, 10.07, 10.08, 10.09, 10.10, 10.11, 10.12, 10.13, 10.14, 10.16, 10.17, 10.18, 10.19, 10.20, 10.21, 10.22, 10.24, 10.25, 10.26, 10.27, 10.28, 10.29, 10.30, 10.31, 10.33, 10.34, 10.35, 10.36, 10.37, 10.38, 10.39, 10.41, 10.42, 10.43, 10.44, 10.45, 10.46, 10.47, 10.49, 10.50, 10.51, 10.52, 10.53, 10.54, 10.55, 10.57, 10.58, 10.59, 10.60, 10.61, 10.62, 10.64, 10.65, 10.66, 10.67, 10.68, 10.69, 10.70, 10.72, 10.73, 10.74, 10.75, 10.76, 10.77, 10.79, 10.80, 10.81, 10.82, 10.83, 10.84, 10.86, 10.87, 10.88, 10.89, 10.90, 10.91, 10.93, 10.94, 10.95, 10.96, 10.97, 10.98, 11.00, 11.01, 11.02, 11.03, 11.04, 11.05, 11.07, 11.08, 11.09, 11.10, 11.11, 11.13, 11.14, 11.15, 11.16, 11.17, 11.18, 11.20, 11.21, 11.22, 11.23, 11.24, 11.26, 11.27, 11.28, 11.29, 11.30, 11.32, 11.33, 11.34, 11.35, 11.36, 11.37, 11.39, 11.40, 11.41, 11.42, 11.43, 11.45, 11.46, 11.47, 11.48, 11.49, 11.51, 11.52, 11.53, 11.54, 11.55, 11.57, 11.58, 11.59, 11.60, 11.61, 11.63, 11.64, 11.65, 11.66, 11.68, 11.69, 11.70, 11.71, 11.72, 11.74, 11.75, 11.76, 11.77, 11.78, 11.80, 11.81, 11.82, 11.83, 11.85, 11.86, 11.87, 11.88, 11.89, 11.91, 11.92, 11.93, 11.94, 11.96, 11.97, 11.98, 11.99, 12.00, 12.02, 12.03, 12.04, 12.05, 12.07, 12.08, 12.09, 12.10, 12.11, 12.13, 12.14, 12.15, 12.16, 12.18, 12.19, 12.20, 12.21, 12.23, 12.24, 12.25, 12.26, 12.28, 12.29, 12.30, 12.31, 12.33, 12.34, 12.35, 12.36, 12.38, 12.39, 12.40, 12.41, 12.43, 12.44, 12.45, 12.46, 12.48, 12.49, 12.50, 12.51, 12.53, 12.54, 12.55, 12.56, 12.58, 12.59, 12.60, 12.61, 12.63, 12.64, 12.65, 12.66, 12.68, 12.69, 12.70, 12.72, 12.73, 12.74, 12.75, 12.77, 12.78, 12.79, 12.80, 12.82, 12.83, 12.84, 12.86, 12.87, 12.88, 12.89, 12.91, 12.92, 12.93, 12.94, 12.96, 12.97, 12.98, 13.00, 13.01, 13.02, 13.03, 13.05, 13.06, 13.07, 13.09, 13.10, 13.11, 13.12, 13.14, 13.15, 13.16, 13.18, 13.19, 13.20, 13.22, 13.23, 13.24, 13.25, 13.27, 13.28, 13.29, 13.31, 13.32, 13.33, 13.35, 13.36, 13.37, 13.38, 13.40, 13.41, 13.42, 13.44, 13.45, 13.46, 13.48, 13.49, 13.50, 13.52, 13.53, 13.54, 13.55, 13.57, 13.58, 13.59, 13.61, 13.62, 13.63, 13.65, 13.66, 13.67, 13.69, 13.70, 13.71, 13.73, 13.74, 13.75, 13.77, 13.78, 13.79, 13.81, 13.82, 13.83, 13.85, 13.86, 13.87, 13.89, 13.90, 13.91, 13.93, 13.94, 13.95, 13.97, 13.98, 13.99, 14.01, 14.02, 14.03, 14.05, 14.06, 14.07, 14.09, 14.10, 14.11, 14.13, 14.14, 14.16, 14.17, 14.18, 14.20, 14.21, 14.22, 14.24, 14.25, 14.26, 14.28, 14.29, 14.30, 14.32, 14.33, 14.35, 14.36, 14.37, 14.39, 14.40, 14.41, 14.43, 14.44, 14.45, 14.47, 14.48, 14.50, 14.51, 14.52, 14.54, 14.55, 14.56, 14.58, 14.59, 14.61, 14.62, 14.63, 14.65, 14.66, 14.67, 14.69, 14.70, 14.72, 14.73, 14.74, 14.76, 14.77, 14.79, 14.80, 14.81, 14.83, 14.84, 14.86, 14.87, 14.88, 14.90, 14.91, 14.93, 14.94, 14.95, 14.97, 14.98, 15.00, 15.01, 15.02, 15.04, 15.05, 15.07, 15.08, 15.09, 15.11, 15.12, 15.14, 15.15, 15.16, 15.18, 15.19, 15.21, 15.22, 15.24, 15.25, 15.26, 15.28, 15.29, 15.31, 15.32, 15.33, 15.35, 15.36, 15.38, 15.39, 15.41, 15.42, 15.43, 15.45, 15.46, 15.48, 15.49, 15.51, 15.52, 15.54, 15.55, 15.56, 15.58, 15.59, 15.61, 15.62, 15.64, 15.65, 15.66, 15.68, 15.69, 15.71, 15.72, 15.74, 15.75, 15.77, 15.78, 15.80, 15.81, 15.82, 15.84, 15.85, 15.87, 15.88, 15.90, 15.91, 15.93, 15.94, 15.96, 15.97, 15.99, 16.00, 16.01, 16.03, 16.04, 16.06, 16.07, 16.09, 16.10, 16.12, 16.13, 16.15, 16.16, 16.18, 16.19, 16.21, 16.22, 16.24, 16.25, 16.27, 16.28, 16.30, 16.31, 16.33, 16.34, 16.36

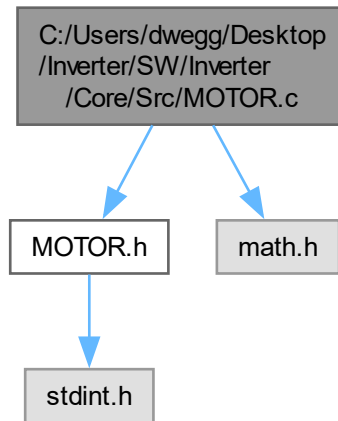
35, 16.37, 16.38, 16.40, 16.41, 16.43, 16.44, 16.46, 16.47, 16.49, 16.50, 16.52, 16.53, 16.↵  
55, 16.56, 16.58, 16.59, 16.61, 16.62, 16.64, 16.66, 16.67, 16.69, 16.70, 16.72, 16.73, 16.↵  
75, 16.76, 16.78, 16.79, 16.81, 16.82, 16.84, 16.85, 16.87, 16.88, 16.90, 16.91, 16.93, 16.↵  
94, 16.96, 16.97, 16.99, 17.01, 17.02, 17.04, 17.05, 17.07, 17.08, 17.10, 17.11, 17.13, 17.↵  
14, 17.16, 17.17, 17.19, 17.21, 17.22, 17.24, 17.25, 17.27, 17.28, 17.30, 17.31, 17.33, 17.↵  
35, 17.36, 17.38, 17.39, 17.41, 17.42, 17.44, 17.45, 17.47, 17.49, 17.50, 17.52, 17.53, 17.↵  
55, 17.56, 17.58, 17.60, 17.61, 17.63, 17.64, 17.66, 17.67, 17.69, 17.71, 17.72, 17.74, 17.↵  
75, 17.77, 17.79, 17.80, 17.82, 17.83, 17.85, 17.86, 17.88, 17.90, 17.91, 17.93, 17.94, 17.↵  
96, 17.98, 17.99, 18.01, 18.02, 18.04, 18.06, 18.07, 18.09, 18.11, 18.12, 18.14, 18.15, 18.↵  
17, 18.19, 18.20, 18.22, 18.23, 18.25, 18.27, 18.28, 18.30, 18.32, 18.33, 18.35, 18.36, 18.↵  
38, 18.40, 18.41, 18.43, 18.45, 18.46, 18.48, 18.49, 18.51, 18.53, 18.54, 18.56, 18.58, 18.↵  
59, 18.61, 18.63, 18.64, 18.66, 18.68, 18.69, 18.71, 18.73, 18.74, 18.76, 18.77, 18.79, 18.↵  
81, 18.82, 18.84, 18.86, 18.87, 18.89, 18.91, 18.92, 18.94, 18.96, 18.97, 18.99, 19.01, 19.↵  
02, 19.04, 19.06, 19.08, 19.09, 19.11, 19.13, 19.14, 19.16, 19.18, 19.19, 19.21, 19.23, 19.↵  
24, 19.26, 19.28, 19.29, 19.31, 19.33, 19.35, 19.36, 19.38, 19.40, 19.41, 19.43, 19.45, 19.↵  
46, 19.48, 19.50, 19.52, 19.53, 19.55, 19.57, 19.58, 19.60, 19.62, 19.64, 19.65, 19.67, 19.↵  
69, 19.70, 19.72, 19.74, 19.76, 19.77, 19.79, 19.81, 19.83, 19.84, 19.86, 19.88, 19.90, 19.↵  
91, 19.93, 19.95, 19.97, 19.98, 20.00, 20.02, 20.04, 20.05, 20.07, 20.09, 20.11, 20.12, 20.↵  
14, 20.16, 20.18, 20.19, 20.21, 20.23, 20.25, 20.26, 20.28, 20.30, 20.32, 20.33, 20.35, 20.↵  
37, 20.39, 20.41, 20.42, 20.44, 20.46, 20.48, 20.49, 20.51, 20.53, 20.55, 20.57, 20.58, 20.↵  
60, 20.62, 20.64, 20.66, 20.67, 20.69, 20.71, 20.73, 20.75, 20.76, 20.78, 20.80, 20.82, 20.↵  
84, 20.85, 20.87, 20.89, 20.91, 20.93, 20.95, 20.96, 20.98, 21.00, 21.02, 21.04, 21.06, 21.↵  
07, 21.09, 21.11, 21.13, 21.15, 21.17, 21.18, 21.20, 21.22, 21.24, 21.26, 21.28, 21.29, 21.↵  
31, 21.33, 21.35, 21.37, 21.39, 21.41, 21.42, 21.44, 21.46, 21.48, 21.50, 21.52, 21.54, 21.↵  
55, 21.57, 21.59, 21.61, 21.63, 21.65, 21.67, 21.69, 21.70, 21.72, 21.74, 21.76, 21.78, 21.↵  
80, 21.82, 21.84, 21.86, 21.87, 21.89, 21.91, 21.93, 21.95, 21.97, 21.99, 22.01, 22.03, 22.↵  
05, 22.06, 22.08, 22.10, 22.12, 22.14, 22.16, 22.18, 22.20, 22.22, 22.24, 22.26, 22.28, 22.↵  
30, 22.31, 22.33, 22.35, 22.37, 22.39, 22.41, 22.43, 22.45, 22.47, 22.49, 22.51, 22.53, 22.↵  
55, 22.57, 22.59, 22.61, 22.63, 22.64, 22.66, 22.68, 22.70, 22.72, 22.74, 22.76, 22.78, 22.↵  
80, 22.82, 22.84, 22.86, 22.88, 22.90, 22.92, 22.94, 22.96, 22.98, 23.00, 23.02, 23.04, 23.↵  
06, 23.08, 23.10, 23.12, 23.14, 23.16, 23.18, 23.20, 23.22, 23.24, 23.26, 23.28, 23.30, 23.↵  
32, 23.34, 23.36, 23.38, 23.40, 23.42, 23.44, 23.46, 23.48, 23.50, 23.52, 23.54, 23.56, 23.↵  
58, 23.60, 23.62, 23.65, 23.67, 23.69, 23.71, 23.73, 23.75, 23.77, 23.79, 23.81, 23.83, 23.↵  
85, 23.87, 23.89, 23.91, 23.93, 23.95, 23.97, 24.00, 24.02, 24.04, 24.06, 24.08, 24.10, 24.↵  
12, 24.14, 24.16, 24.18, 24.20, 24.22, 24.25, 24.27, 24.29, 24.31, 24.33, 24.35, 24.37, 24.↵  
39, 24.41, 24.43, 24.46, 24.48, 24.50, 24.52, 24.54, 24.56, 24.58, 24.60, 24.63, 24.65, 24.↵  
67, 24.69, 24.71, 24.73, 24.75, 24.78, 24.80, 24.82, 24.84, 24.86, 24.88, 24.90, 24.93, 24.↵  
95, 24.97, 24.99, 25.01, 25.03, 25.06, 25.08, 25.10, 25.12, 25.14, 25.16, 25.19, 25.21, 25.↵  
23, 25.25, 25.27, 25.30, 25.32, 25.34, 25.36, 25.38, 25.41, 25.43, 25.45, 25.47, 25.49, 25.↵  
52, 25.54, 25.56, 25.58, 25.60, 25.63, 25.65, 25.67, 25.69, 25.72, 25.74, 25.76, 25.78, 25.↵  
81, 25.83, 25.85, 25.87, 25.89, 25.92, 25.94, 25.96, 25.98, 26.01, 26.03, 26.05, 26.08, 26.↵  
10, 26.12, 26.14, 26.17, 26.19, 26.21, 26.23, 26.26, 26.28, 26.30, 26.33, 26.35, 26.37, 26.↵  
39, 26.42, 26.44, 26.46, 26.49, 26.51, 26.53, 26.56, 26.58, 26.60, 26.63, 26.65, 26.67, 26.↵  
69, 26.72, 26.74, 26.76, 26.79, 26.81, 26.83, 26.86, 26.88, 26.90, 26.93, 26.95, 26.98, 27.↵  
00, 27.02, 27.05, 27.07, 27.09, 27.12, 27.14, 27.16, 27.19, 27.21, 27.24, 27.26, 27.28, 27.↵  
31, 27.33, 27.35, 27.38, 27.40, 27.43, 27.45, 27.47, 27.50, 27.52, 27.55, 27.57, 27.59, 27.↵  
62, 27.64, 27.67, 27.69, 27.72, 27.74, 27.76, 27.79, 27.81, 27.84, 27.86, 27.89, 27.91, 27.↵  
93, 27.96, 27.98, 28.01, 28.03, 28.06, 28.08, 28.11, 28.13, 28.16, 28.18, 28.21, 28.23, 28.↵  
26, 28.28, 28.30, 28.33, 28.35, 28.38, 28.40, 28.43, 28.45, 28.48, 28.50, 28.53, 28.55, 28.↵  
58, 28.60, 28.63, 28.66, 28.68, 28.71, 28.73, 28.76, 28.78, 28.81, 28.83, 28.86, 28.88, 28.↵  
91, 28.93, 28.96, 28.99, 29.01, 29.04, 29.06, 29.09, 29.11, 29.14, 29.17, 29.19, 29.22, 29.↵  
24, 29.27, 29.29, 29.32, 29.35, 29.37, 29.40, 29.42, 29.45, 29.48, 29.50, 29.53, 29.55, 29.↵  
58, 29.61, 29.63, 29.66, 29.69, 29.71, 29.74, 29.76, 29.79, 29.82, 29.84, 29.87, 29.90, 29.↵  
92, 29.95, 29.98, 30.00, 30.03, 30.06, 30.08, 30.11, 30.14, 30.16, 30.19, 30.22, 30.24, 30.↵  
27, 30.30, 30.33, 30.35, 30.38, 30.41, 30.43, 30.46, 30.49, 30.52, 30.54, 30.57, 30.60, 30.↵  
62, 30.65, 30.68, 30.71, 30.73, 30.76, 30.79, 30.82, 30.84, 30.87, 30.90, 30.93, 30.96, 30.↵  
98, 31.01, 31.04, 31.07, 31.09, 31.12, 31.15, 31.18, 31.21, 31.23, 31.26, 31.29, 31.32, 31.↵  
35, 31.37, 31.40, 31.43, 31.46, 31.49, 31.52, 31.54, 31.57, 31.60, 31.63, 31.66, 31.69, 31.↵

72, 31.74, 31.77, 31.80, 31.83, 31.86, 31.89, 31.92, 31.95, 31.97, 32.00, 32.03, 32.06, 32.↵  
09, 32.12, 32.15, 32.18, 32.21, 32.24, 32.27, 32.29, 32.32, 32.35, 32.38, 32.41, 32.44, 32.↵  
47, 32.50, 32.53, 32.56, 32.59, 32.62, 32.65, 32.68, 32.71, 32.74, 32.77, 32.80, 32.83, 32.↵  
86, 32.89, 32.92, 32.95, 32.98, 33.01, 33.04, 33.07, 33.10, 33.13, 33.16, 33.19, 33.22, 33.↵  
25, 33.28, 33.31, 33.34, 33.37, 33.40, 33.43, 33.46, 33.49, 33.53, 33.56, 33.59, 33.62, 33.↵  
65, 33.68, 33.71, 33.74, 33.77, 33.80, 33.84, 33.87, 33.90, 33.93, 33.96, 33.99, 34.02, 34.↵  
05, 34.09, 34.12, 34.15, 34.18, 34.21, 34.24, 34.28, 34.31, 34.34, 34.37, 34.40, 34.43, 34.↵  
47, 34.50, 34.53, 34.56, 34.59, 34.63, 34.66, 34.69, 34.72, 34.76, 34.79, 34.82, 34.85, 34.↵  
89, 34.92, 34.95, 34.98, 35.02, 35.05, 35.08, 35.11, 35.15, 35.18, 35.21, 35.25, 35.28, 35.↵  
31, 35.35, 35.38, 35.41, 35.44, 35.48, 35.51, 35.54, 35.58, 35.61, 35.65, 35.68, 35.71, 35.↵  
75, 35.78, 35.81, 35.85, 35.88, 35.91, 35.95, 35.98, 36.02, 36.05, 36.08, 36.12, 36.15, 36.↵  
19, 36.22, 36.26, 36.29, 36.33, 36.36, 36.39, 36.43, 36.46, 36.50, 36.53, 36.57, 36.60, 36.↵  
64, 36.67, 36.71, 36.74, 36.78, 36.81, 36.85, 36.88, 36.92, 36.95, 36.99, 37.02, 37.06, 37.↵  
09, 37.13, 37.17, 37.20, 37.24, 37.27, 37.31, 37.34, 37.38, 37.42, 37.45, 37.49, 37.52, 37.↵  
56, 37.60, 37.63, 37.67, 37.71, 37.74, 37.78, 37.82, 37.85, 37.89, 37.93, 37.96, 38.00, 38.↵  
04, 38.07, 38.11, 38.15, 38.18, 38.22, 38.26, 38.30, 38.33, 38.37, 38.41, 38.44, 38.48, 38.↵  
52, 38.56, 38.60, 38.63, 38.67, 38.71, 38.75, 38.78, 38.82, 38.86, 38.90, 38.94, 38.97, 39.↵  
01, 39.05, 39.09, 39.13, 39.17, 39.21, 39.24, 39.28, 39.32, 39.36, 39.40, 39.44, 39.48, 39.↵  
52, 39.56, 39.59, 39.63, 39.67, 39.71, 39.75, 39.79, 39.83, 39.87, 39.91, 39.95, 39.99, 40.↵  
03, 40.07, 40.11, 40.15, 40.19, 40.23, 40.27, 40.31, 40.35, 40.39, 40.43, 40.47, 40.51, 40.↵  
55, 40.59, 40.64, 40.68, 40.72, 40.76, 40.80, 40.84, 40.88, 40.92, 40.96, 41.01, 41.05, 41.↵  
09, 41.13, 41.17, 41.21, 41.26, 41.30, 41.34, 41.38, 41.42, 41.47, 41.51, 41.55, 41.59, 41.↵  
64, 41.68, 41.72, 41.76, 41.81, 41.85, 41.89, 41.93, 41.98, 42.02, 42.06, 42.11, 42.15, 42.↵  
19, 42.24, 42.28, 42.32, 42.37, 42.41, 42.46, 42.50, 42.54, 42.59, 42.63, 42.68, 42.72, 42.↵  
76, 42.81, 42.85, 42.90, 42.94, 42.99, 43.03, 43.08, 43.12, 43.17, 43.21, 43.26, 43.30, 43.↵  
35, 43.39, 43.44, 43.48, 43.53, 43.58, 43.62, 43.67, 43.71, 43.76, 43.81, 43.85, 43.90, 43.↵  
94, 43.99, 44.04, 44.08, 44.13, 44.18, 44.23, 44.27, 44.32, 44.37, 44.41, 44.46, 44.51, 44.↵  
56, 44.60, 44.65, 44.70, 44.75, 44.80, 44.84, 44.89, 44.94, 44.99, 45.04, 45.08, 45.13, 45.↵  
18, 45.23, 45.28, 45.33, 45.38, 45.43, 45.48, 45.53, 45.57, 45.62, 45.67, 45.72, 45.77, 45.↵  
82, 45.87, 45.92, 45.97, 46.02, 46.07, 46.12, 46.17, 46.23, 46.28, 46.33, 46.38, 46.43, 46.↵  
48, 46.53, 46.58, 46.63, 46.69, 46.74, 46.79, 46.84, 46.89, 46.95, 47.00, 47.05, 47.10, 47.↵  
15, 47.21, 47.26, 47.31, 47.37, 47.42, 47.47, 47.52, 47.58, 47.63, 47.69, 47.74, 47.79, 47.↵  
85, 47.90, 47.95, 48.01, 48.06, 48.12, 48.17, 48.23, 48.28, 48.34, 48.39, 48.45, 48.50, 48.↵  
56, 48.61, 48.67, 48.72, 48.78, 48.84, 48.89, 48.95, 49.00, 49.06, 49.12, 49.17, 49.23, 49.↵  
29, 49.34, 49.40, 49.46, 49.52, 49.57, 49.63, 49.69, 49.75, 49.80, 49.86, 49.92, 49.98, 50.↵  
04, 50.10, 50.16, 50.21, 50.27, 50.33, 50.39, 50.45, 50.51, 50.57, 50.63, 50.69, 50.75, 50.↵  
81, 50.87, 50.93, 50.99, 51.05, 51.11, 51.17, 51.24, 51.30, 51.36, 51.42, 51.48, 51.54, 51.↵  
61, 51.67, 51.73, 51.79, 51.86, 51.92, 51.98, 52.04, 52.11, 52.17, 52.23, 52.30, 52.36, 52.↵  
43, 52.49, 52.55, 52.62, 52.68, 52.75, 52.81, 52.88, 52.94, 53.01, 53.07, 53.14, 53.21, 53.↵  
27, 53.34, 53.40, 53.47, 53.54, 53.60, 53.67, 53.74, 53.80, 53.87, 53.94, 54.01, 54.08, 54.↵  
14, 54.21, 54.28, 54.35, 54.42, 54.49, 54.56, 54.63, 54.70, 54.76, 54.83, 54.90, 54.98, 55.↵  
05, 55.12, 55.19, 55.26, 55.33, 55.40, 55.47, 55.54, 55.62, 55.69, 55.76, 55.83, 55.91, 55.↵  
98, 56.05, 56.12, 56.20, 56.27, 56.35, 56.42, 56.49, 56.57, 56.64, 56.72, 56.79, 56.87, 56.↵  
94, 57.02, 57.10, 57.17, 57.25, 57.32, 57.40, 57.48, 57.56, 57.63, 57.71, 57.79, 57.87, 57.↵  
94, 58.02, 58.10, 58.18, 58.26, 58.34, 58.42, 58.50, 58.58, 58.66, 58.74, 58.82, 58.90, 58.↵  
98, 59.06, 59.15, 59.23, 59.31, 59.39, 59.48, 59.56, 59.64, 59.72, 59.81, 59.89, 59.98, 60.↵  
06, 60.15, 60.23, 60.32, 60.40, 60.49, 60.57, 60.66, 60.75, 60.83, 60.92, 61.01, 61.10, 61.↵  
18, 61.27, 61.36, 61.45, 61.54, 61.63, 61.72, 61.81, 61.90, 61.99, 62.08, 62.17, 62.26, 62.↵  
35, 62.44, 62.54, 62.63, 62.72, 62.82, 62.91, 63.00, 63.10, 63.19, 63.29, 63.38, 63.48, 63.↵  
57, 63.67, 63.76, 63.86, 63.96, 64.06, 64.15, 64.25, 64.35, 64.45, 64.55, 64.65, 64.75, 64.↵  
85, 64.95, 65.05, 65.15, 65.25, 65.35, 65.46, 65.56, 65.66, 65.76, 65.87, 65.97, 66.08, 66.↵  
18, 66.29, 66.39, 66.50, 66.61, 66.71, 66.82, 66.93, 67.03, 67.14, 67.25, 67.36, 67.47, 67.↵  
58, 67.69, 67.80, 67.91, 68.03, 68.14, 68.25, 68.36, 68.48, 68.59, 68.71, 68.82, 68.94, 69.↵  
05, 69.17, 69.29, 69.40, 69.52, 69.64, 69.76, 69.88, 70.00, 70.12, 70.24, 70.36, 70.48, 70.↵  
60, 70.73, 70.85, 70.97, 71.10, 71.22, 71.35, 71.47, 71.60, 71.73, 71.86, 71.98, 72.11, 72.↵  
24, 72.37, 72.50, 72.63, 72.76, 72.90, 73.03, 73.16, 73.30, 73.43, 73.57, 73.70, 73.84, 73.↵  
98, 74.11, 74.25, 74.39, 74.53, 74.67, 74.81, 74.95, 75.10, 75.24, 75.38, 75.53, 75.67, 75.↵  
82, 75.97, 76.11, 76.26, 76.41, 76.56, 76.71, 76.86, 77.01, 77.17, 77.32, 77.47, 77.63, 77.↵

#### 4.34 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/MOTOR.c File Reference

Generated by Doxygen

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



## Functions

- void `precalculate_motor_constants` (`MotorParameters *motor`)  
*Precomputes the constants for a motor and updates the `MotorParameters` structure.*
- int `check_motor_parameters` (`MotorParameters *motor`, float Ts)  
*Perform a parameter check and correct possible errors.*

## Variables

- `MotorParameters motor_left`  
*Left motor parameters.*
- `MotorParameters motor_right`  
*Right motor parameters.*

### 4.34.1 Detailed Description

Source file for motor parameters.

#### Attention

Copyright (c) 2024 David Redondo (@dweggg on 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>



## 4.34.2 Function Documentation

### 4.34.2.1 check\_motor\_parameters()

```
int check_motor_parameters (
    MotorParameters * motor,
    float Ts )
```

Perform a parameter check and correct possible errors.

#### Parameters

in	<i>motor</i>	Pointer to the <a href="#">MotorParameters</a> struct.
----	--------------	--

#### Return values

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

Here is the caller graph for this function:



### 4.34.2.2 precalculate\_motor\_constants()

```
void precalculate_motor_constants (
    MotorParameters * motor )
```

Precomputes the constants for a motor and updates the [MotorParameters](#) structure.

#### Parameters

<i>motor</i>	[in, out] Pointer to the motor parameters structure
--------------	---

Here is the caller graph for this function:



### 4.34.3 Variable Documentation

#### 4.34.3.1 motor\_left

`MotorParameters` `motor_left`

##### Initial value:

```
= {  
    .Ld = 0.00291F,  
    .Lq = 0.00291F,  
    .Rs = 1.95F,  
    .lambda = 0.13391F,  
    .pp = 4,  
    .J = 0.00093F,  
    .b = 0.632653F,  
    .torqueMax = 10.0F,  
    .dTorqueMax = 1.0F,  
    .speedMax_RPM = 8500.0F,  
    .iMax = 60.0F,  
    .vDCMax = 450.0F  
}
```

Left motor parameters.

#### 4.34.3.2 motor\_right

`MotorParameters` `motor_right`

##### Initial value:

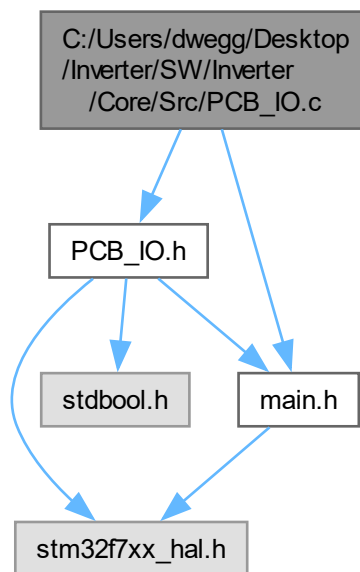
```
= {  
    .Ld = 0.00291F,  
    .Lq = 0.00291F,  
    .Rs = 1.95F,  
    .lambda = 0.13391F,  
    .pp = 4,  
    .J = 0.00093F,  
    .b = 0.632653F,  
    .torqueMax = 10.0F,  
    .dTorqueMax = 1.0F,  
    .speedMax_RPM = 8500.0F,  
    .iMax = 60.0F,  
    .vDCMax = 450.0F  
}
```

Right motor parameters.

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

This file provides functions for handling GPIOs.

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



### Functions

- void [handle\\_LED](#) (LED \*led, uint32\_t ms\_counter)  
*LED handler function.*
- void [handle\\_direction](#) (volatile int8\_t \*dir\_left, volatile int8\_t \*dir\_right)  
*Handles the direction of the motors.*
- void [enable\\_inverters](#) (volatile bool enableSW\_left, volatile bool enableSW\_right, volatile bool \*enable\_left, volatile bool \*enable\_right)  
*Handles the direction of the motors and enables/disables the inverters.*

### 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 [ledError](#) = { .port = [LED\\_ERR\\_GPIO\\_Port](#), .pin = [LED\\_ERR\\_Pin](#), .mode = [LED\\_MODE\\_OFF](#) }

### 4.35.1 Detailed Description

This file provides functions for handling GPIOs.

#### 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>

### 4.35.2 Function Documentation

#### 4.35.2.1 enable\_inverters()

```
void enable_inverters (
    volatile bool enableSW_left,
    volatile bool enableSW_right,
    volatile bool * enable_left,
    volatile bool * enable_right )
```

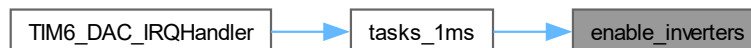
Handles the direction of the motors and enables/disables the inverters.

This function reads the state of the shutdown chain (SC or SDC) and enables/disables the inverters based on that and an external software enable bool.

#### Parameters

in	<i>enableSW_left</i>	The software enable state for the left inverter.
in	<i>enableSW_right</i>	The software enable state for the right inverter.
out	<i>enable_left</i>	Output parameter for the left inverter's enable state.
out	<i>enable_right</i>	Output parameter for the right inverter's enable state.

Here is the caller graph for this function:



#### 4.35.2.2 handle\_direction()

```
void handle_direction (
```

```
volatile int8_t * dir_left,  
volatile int8_t * dir_right )
```

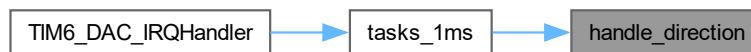
Handles the direction of the motors.

This function reads the state of the DIR switch and updates the directions of both the left and right motors. If one motor is set to rotate clockwise (CW), the other one is set to rotate counterclockwise (CCW), and vice versa.

#### Parameters

<i>dir_left</i>	Pointer to the direction parameter in the left inverter structure.
<i>dir_right</i>	Pointer to the direction parameter in the right inverter structure.

Here is the caller graph for this function:



#### 4.35.2.3 handle\_LED()

```
void handle_LED (  
    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 LED structure.
<i>ms_counter</i>	Current millisecond counter.

Here is the caller graph for this function:



### 4.35.3 Variable Documentation

#### 4.35.3.1 led\_left

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

#### 4.35.3.2 led\_right

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

#### 4.35.3.3 ledError

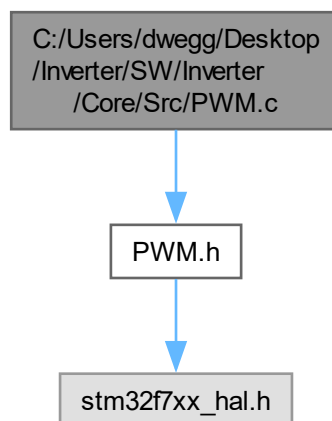
```
LED ledError = { .port = LED_ERR_GPIO_Port, .pin = LED_ERR_Pin, .mode = LED_MODE_OFF }
```

## 4.36 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 [update\\_PWM](#) (TIM\_HandleTypeDef \*htim, [Duties](#) duties)  
*Set PWM duty cycles.*

### 4.36.1 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>

### 4.36.2 Function Documentation

#### 4.36.2.1 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.
-------------	---

#### 4.36.2.2 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.
-------------	---

#### 4.36.2.3 update\_PWM()

```
void update_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>	<a href="#">Duties</a> structure containing duty cycle values.

Here is the caller graph for this function:



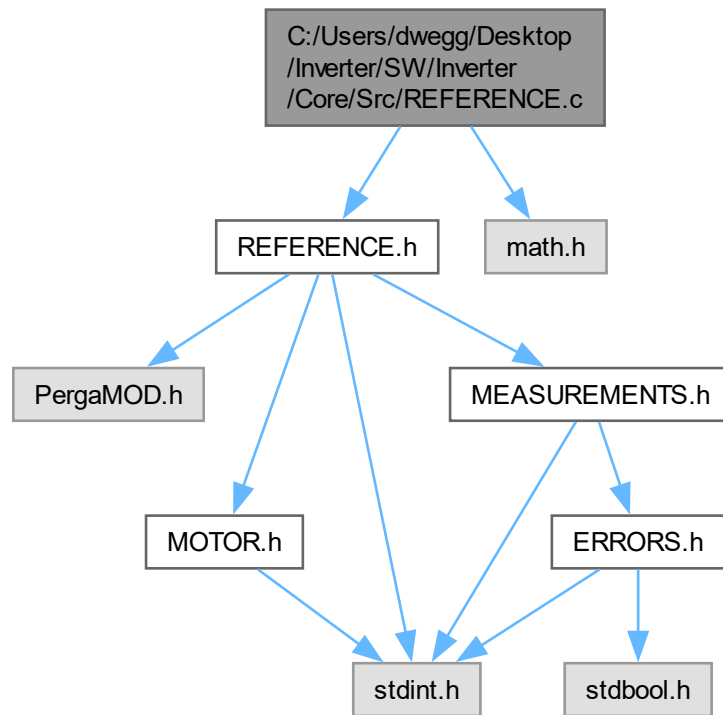
### 4.37 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/↵ REFERENCE.c File Reference

Source file for torque reference handling.

```
#include "REFERENCE.h"  
#include <math.h>
```



Include dependency graph for REFERENCE.c:



## Functions

- float [handle\\_torqueRef](#) (float torqueRefIn, int8\_t direction, float torqueMax, float speedMaxRPM, float speedMeas, volatile pi\_struct \*loopSpeed)  
*Handles torque control based on the reference torque, direction, maximum torque, maximum speed, measured speed, maximum torque rate of change, speed control loop parameters, and sampling time.*
- float [set\\_torque\\_direction](#) (float torqueRefIn, int8\_t direction)  
*Set torque direction based on inverter direction.*
- float [saturate\\_symmetric](#) (float ref, float max)  
*Symmetrically saturate a reference value.*
- float [limit\\_torque\\_to\\_prevent\\_overspeed](#) (float speedMaxRPM, float speedMeas, float torqueRefIn, volatile pi\_struct \*loopSpeed)  
*Speed loop acts as a torque saturation, reducing torque in order to limit the maximum speed.*
- float [calculate\\_derated\\_current](#) (float temperature, float tempStart, float tempMax, float iMax)  
*Calculate derated current based on temperature thresholds. It implements a simple linear derating from tempStart to tempMax.*
- float [derate\\_current\\_reference](#) (float tempMotor, float tempInverter, float iMax)  
*Derate the current reference based on both motor and inverter temperatures.*

## Variables

- float [torqueRefIn\\_left](#) = 0.0F
- float [torqueRefIn\\_right](#) = 0.0F

### 4.37.1 Detailed Description

Source file for torque reference handling.

#### 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>

### 4.37.2 Function Documentation

#### 4.37.2.1 calculate\_derated\_current()

```
float calculate_derated_current (
    float temperature,
    float tempStart,
    float tempMax,
    float iMax )
```

Calculate derated current based on temperature thresholds. It implements a simple linear derating from tempStart to tempMax.

#### Parameters

in	<i>temperature</i>	The current temperature.
in	<i>tempStart</i>	The temperature at which derating starts.
in	<i>tempMax</i>	The temperature at which the current is fully derated to 0.
in	<i>iMax</i>	The maximum current.

#### Returns

The derated current.

Here is the caller graph for this function:



#### 4.37.2.2 derate\_current\_reference()

```
float derate_current_reference (
    float tempMotor,
```

```
float tempInverter,
float iMax )
```

Derate the current reference based on both motor and inverter temperatures.

#### Parameters

in	<i>tempMotor</i>	The motor temperature.
in	<i>tempInverter</i>	The inverter temperature.
in	<i>iMax</i>	The maximum current.

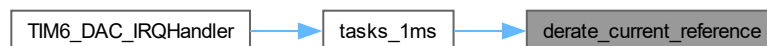
#### Returns

The derated current reference.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.37.2.3 handle\_torqueRef()

```
float handle_torqueRef (
    float torqueRefIn,
    int8_t direction,
    float torqueMax,
    float speedMaxRPM,
    float speedMeas,
    volatile pi_struct * loopSpeed )
```

Handles torque control based on the reference torque, direction, maximum torque, maximum speed, measured speed, maximum torque rate of change, speed control loop parameters, and sampling time.

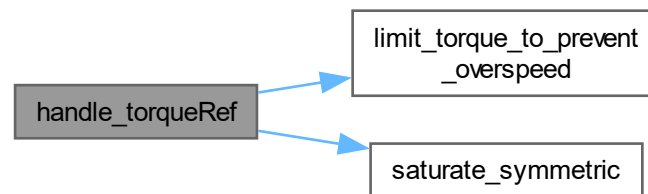
## Parameters

<i>torqueRefIn</i>	Input reference torque.
<i>direction</i>	Direction of torque (1 for positive torque, -1 for negative torque).
<i>torqueMax</i>	Maximum allowable torque.
<i>speedMaxRPM</i>	Maximum allowable speed in RPM.
<i>speedMeas</i>	Measured speed.
<i>loopSpeed</i>	Speed control loop parameters.

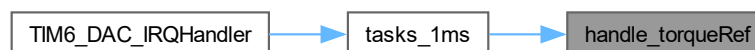
## Returns

The output torque after handling direction, saturation, and rate limiting.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.37.2.4 limit\_torque\_to\_prevent\_overspeed()

```

float limit_torque_to_prevent_overspeed (
    float speedMaxRPM,
    float speedMeas,
    float torqueRefIn,
    volatile pi_struct * loopSpeed )
  
```

Speed loop acts as a torque saturation, reducing torque in order to limit the maximum speed.

**Parameters**

in	<i>speedMaxRPM</i>	The maximum speed value in RPM.
in	<i>speedMeas</i>	The measured speed value in RPM.
in	<i>torqueRefIn</i>	The torque reference value before this saturation.
in	<i>loopSpeed</i>	Pointer to the speed PI controller structure.

**Returns**

torqueRefOut The limited torque reference value after this saturation.

Here is the caller graph for this function:

**4.37.2.5 saturate\_symmetric()**

```
float saturate_symmetric (
    float ref,
    float max )
```

Symmetrically saturate a reference value.

This function symmetrically saturates a reference value based on the maximum allowed value. If the reference value exceeds the maximum allowed value, it is saturated to the maximum value. If the reference value is less than the negative of the maximum allowed value, it is saturated to the negative of the maximum value.

**Parameters**

in	<i>ref</i>	The reference value to saturate.
in	<i>max</i>	The maximum allowed value for saturation.

**Returns**

The saturated reference value.

Here is the caller graph for this function:



#### 4.37.2.6 set\_torque\_direction()

```
float set_torque_direction (
    float torqueRefIn,
    int8_t direction )
```

Set torque direction based on inverter direction.

This function adjusts the torque reference based on the desired direction. If the motor is set to rotate counterclockwise (CCW), positive torque represents traction, negative is braking. If the motor is set to rotate clockwise (CW), negative torque represents traction, positive is braking.

##### Parameters

in	<i>torque↵ RefIn</i>	The torque reference value to adjust.
in	<i>direction</i>	Pointer to the direction of the inverter (1 for CW, -1 for CCW).

##### Returns

torqueRefOut The adjusted torque reference value.

### 4.37.3 Variable Documentation

#### 4.37.3.1 torqueRefIn\_left

```
float torqueRefIn_left = 0.0F
```

#### 4.37.3.2 torqueRefIn\_right

```
float torqueRefIn_right = 0.0F
```

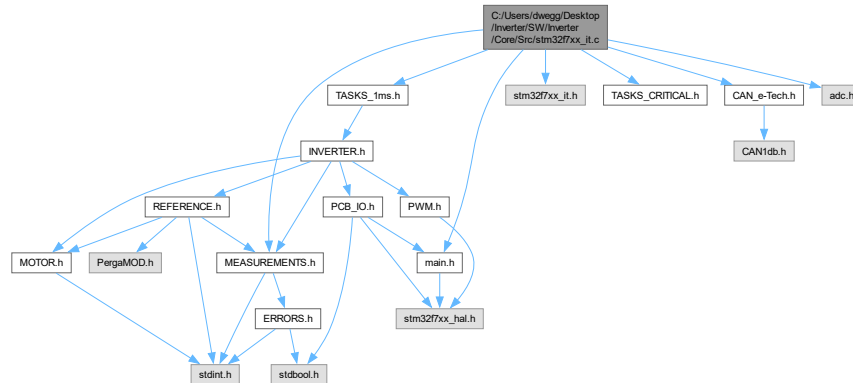
## 4.38 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/stm32f7xx↵ \_it.c File Reference

Interrupt Service Routines.

```
#include "main.h"
#include "stm32f7xx_it.h"
#include "TASKS_lms.h"
#include "TASKS_CRITICAL.h"
#include "CAN_e-Tech.h"
#include "adc.h"
```

```
#include "MEASUREMENTS.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)  
*This function handles CAN1 RX0 interrupts.*
- 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\\_Stream0\\_IRQHandler](#) (void)  
*This function handles DMA2 stream0 global interrupt.*
- void [DMA2\\_Stream1\\_IRQHandler](#) (void)  
*This function handles DMA2 stream1 global interrupt.*
- void [DMA2\\_Stream2\\_IRQHandler](#) (void)  
*This function handles DMA2 stream2 global interrupt.*

## Variables

- DMA\_HandleTypeDef [hdma\\_adc1](#)
- DMA\_HandleTypeDef [hdma\\_adc2](#)
- DMA\_HandleTypeDef [hdma\\_adc3](#)
- CAN\_HandleTypeDef [hcan1](#)
- DAC\_HandleTypeDef [hdac](#)
- TIM\_HandleTypeDef [htim1](#)
- TIM\_HandleTypeDef [htim6](#)

### 4.38.1 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.

### 4.38.2 Function Documentation

#### 4.38.2.1 BusFault\_Handler()

```
void BusFault_Handler (  
    void )
```

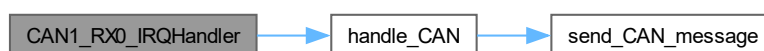
This function handles Pre-fetch fault, memory access fault.

#### 4.38.2.2 CAN1\_RX0\_IRQHandler()

```
void CAN1_RX0_IRQHandler (  
    void )
```

This function handles CAN1 RX0 interrupts.

Here is the call graph for this function:





#### 4.38.2.3 DebugMon\_Handler()

```
void DebugMon_Handler (  
    void )
```

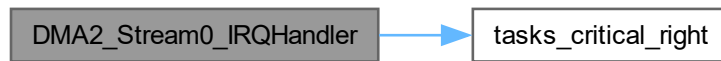
This function handles Debug monitor.

#### 4.38.2.4 DMA2\_Stream0\_IRQHandler()

```
void DMA2_Stream0_IRQHandler (  
    void )
```

This function handles DMA2 stream0 global interrupt.

Here is the call graph for this function:



#### 4.38.2.5 DMA2\_Stream1\_IRQHandler()

```
void DMA2_Stream1_IRQHandler (  
    void )
```

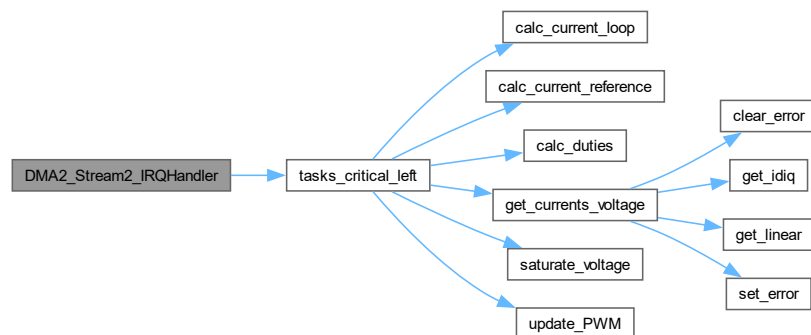
This function handles DMA2 stream1 global interrupt.

#### 4.38.2.6 DMA2\_Stream2\_IRQHandler()

```
void DMA2_Stream2_IRQHandler (  
    void )
```

This function handles DMA2 stream2 global interrupt.

Here is the call graph for this function:



#### 4.38.2.7 HardFault\_Handler()

```
void HardFault_Handler (
    void )
```

This function handles Hard fault interrupt.

#### 4.38.2.8 MemManage\_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

#### 4.38.2.9 NMI\_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

#### 4.38.2.10 PendSV\_Handler()

```
void PendSV_Handler (
    void )
```

This function handles Pendable request for system service.

#### 4.38.2.11 SVC\_Handler()

```
void SVC_Handler (
    void )
```

This function handles System service call via SWI instruction.

#### 4.38.2.12 SysTick\_Handler()

```
void SysTick_Handler (
    void )
```

This function handles System tick timer.

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

```
void TIM1_UP_TIM10_IRQHandler (
    void )
```

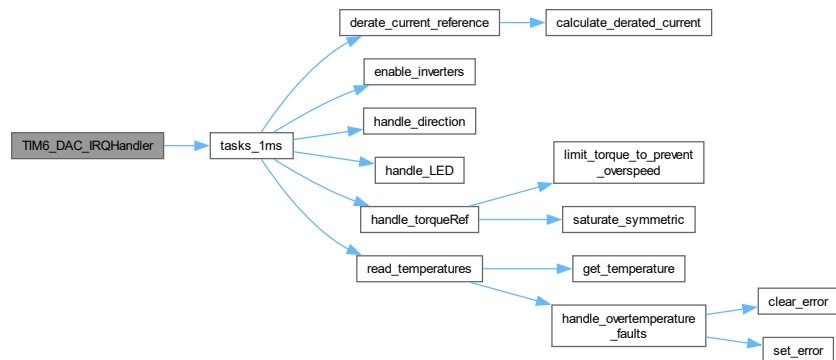
This function handles TIM1 update interrupt and TIM10 global interrupt.

#### 4.38.2.14 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:



#### 4.38.2.15 UsageFault\_Handler()

```
void UsageFault_Handler (
    void )
```

This function handles Undefined instruction or illegal state.

### 4.38.3 Variable Documentation

#### 4.38.3.1 hcan1

```
CAN_HandleTypeDef hcan1 [extern]
```

#### 4.38.3.2 hdac

```
DAC_HandleTypeDef hdac [extern]
```

#### 4.38.3.3 hdma\_adc1

```
DMA_HandleTypeDef hdma_adc1 [extern]
```

#### 4.38.3.4 hdma\_adc2

```
DMA_HandleTypeDef hdma_adc2 [extern]
```

#### 4.38.3.5 hdma\_adc3

```
DMA_HandleTypeDef hdma_adc3 [extern]
```

#### 4.38.3.6 htim1

```
TIM_HandleTypeDef htim1 [extern]
```

#### 4.38.3.7 htim6

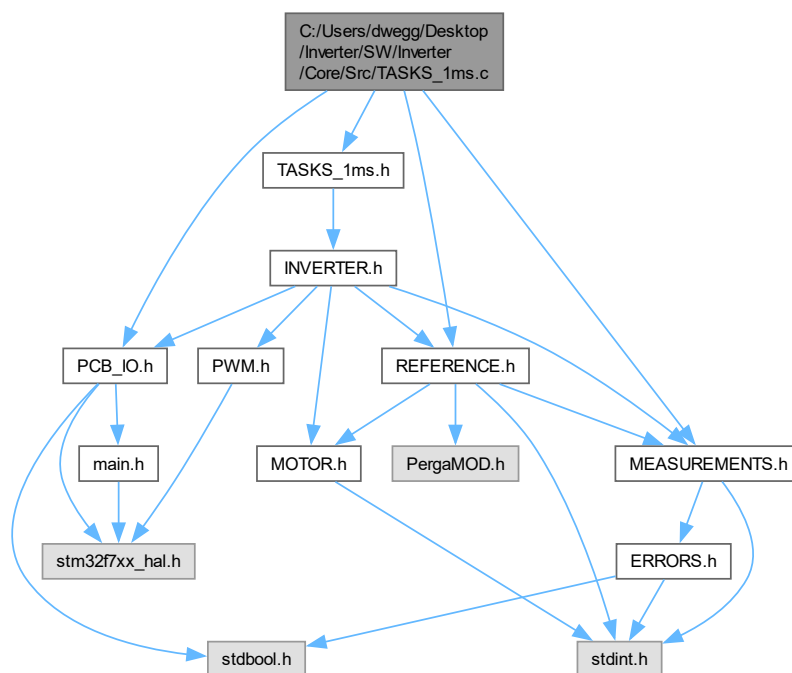
```
TIM_HandleTypeDef htim6 [extern]
```

### 4.39 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 "MEASUREMENTS.h"
#include "REFERENCE.h"
```

Include dependency graph for TASKS\_1ms.c:



## Functions

- void `tasks_1ms` (void)  
*Function to be executed every 1ms.*
- void `read_temperatures` (void)  
*Function to read temperatures and handle overtemperature faults.*
- void `handle_overtemperature_faults` (volatile `InverterStruct` \*inv)  
*Function to handle overtemperature faults.*

### 4.39.1 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>

### 4.39.2 Function Documentation

#### 4.39.2.1 `handle_overtemperature_faults()`

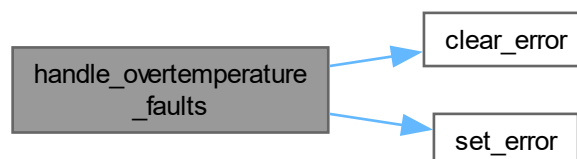
```
void handle_overtemperature_faults (  
    volatile InverterStruct * inv )
```

Function to handle overtemperature faults.

#### Parameters

<code>in, out</code>	<code>inv</code>	Pointer to the <code>InverterStruct</code> structure.
----------------------	------------------	---

Here is the call graph for this function:



Here is the caller graph for this function:

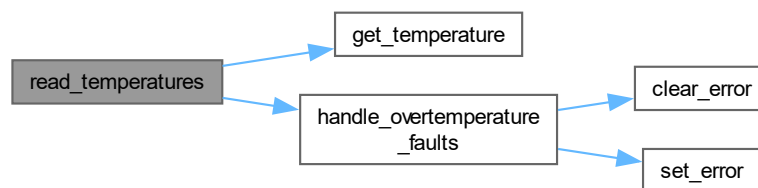


#### 4.39.2.2 read\_temperatures()

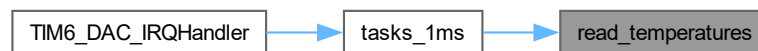
```
void read_temperatures (  
    void )
```

Function to read temperatures and handle overtemperature faults.

Here is the call graph for this function:



Here is the caller graph for this function:

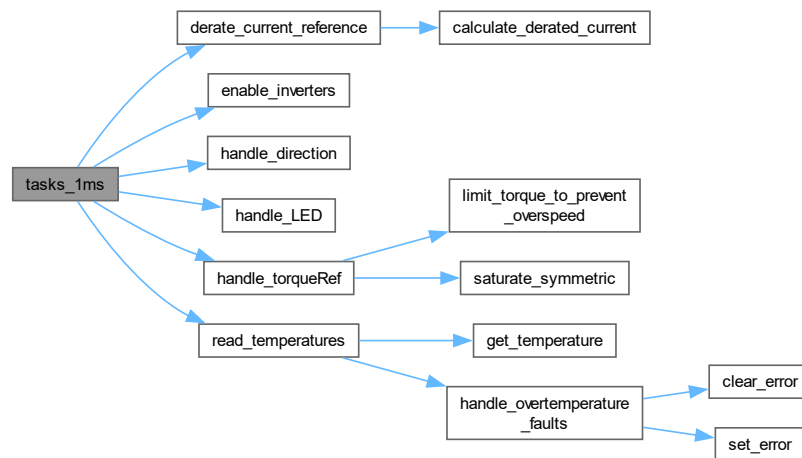


#### 4.39.2.3 tasks\_1ms()

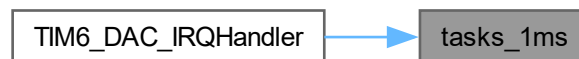
```
void tasks_1ms (  
    void )
```

Function to be executed every 1ms.

This function is called by the TIM6 IRQ handler every millisecond. It increments the millisecond counter and executes all the low priority tasks. Here is the call graph for this function:



Here is the caller graph for this function:



## 4.40 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/TASKS\_CRITICAL.c File Reference

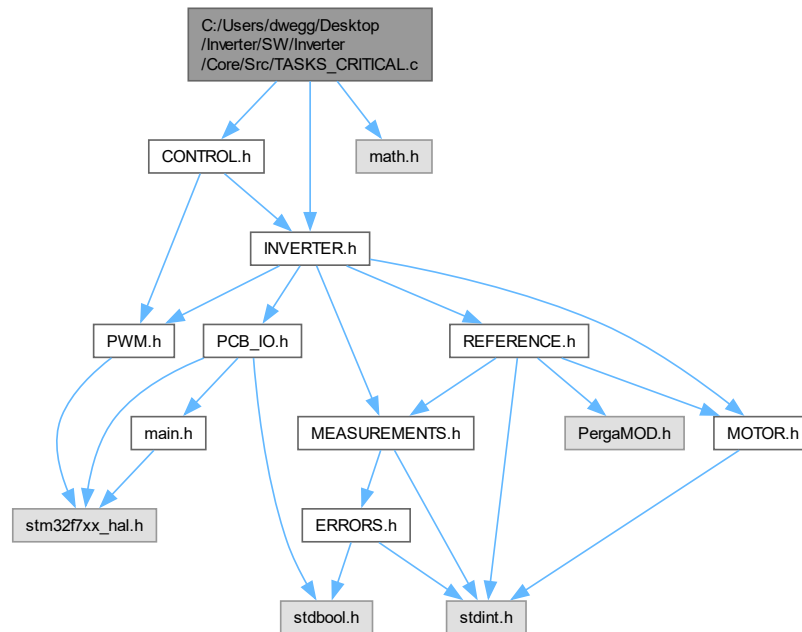
This file contains functions executed in each PWM timer interruption.

```

#include "CONTROL.h"
#include "INVERTER.h"
#include <math.h>

```

Include dependency graph for TASKS\_CRITICAL.c:



## Functions

- void `tasks_critical_left` (void)  
*Function to be executed every TS.*
- void `tasks_critical_right` (void)  
*Function to be executed every TS.*

## Variables

- uint32\_t `start_ticks` = 0
- uint32\_t `elapsed_ticks` = 0
- angle\_struct `angle_left`
- rampa\_struct `freqRamp_left`

### 4.40.1 Detailed Description

This file contains functions executed in each PWM timer interruption.

#### 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>



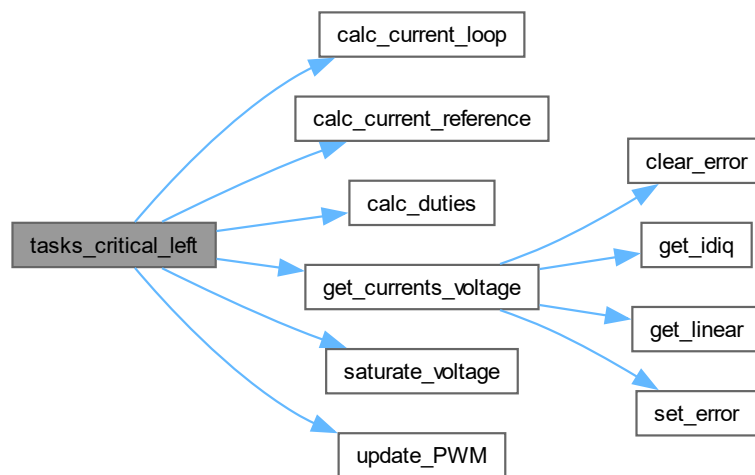
## 4.40.2 Function Documentation

### 4.40.2.1 tasks\_critical\_left()

```
void tasks_critical_left (  
    void )
```

Function to be executed every TS.

This function is called by the TIM1 trigger handler every TS. Here is the call graph for this function:



Here is the caller graph for this function:

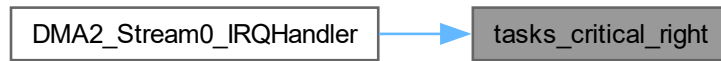


### 4.40.2.2 tasks\_critical\_right()

```
void tasks_critical_right (  
    void )
```

Function to be executed every TS.

This function is called by the TIM8 trigger handler every TS. Here is the caller graph for this function:



### 4.40.3 Variable Documentation

#### 4.40.3.1 angle\_left

```
angle_struct angle_left
```

**Initial value:**

```
= {  
    .freq = 0.0F,  
    .Ts = TS,  
}
```

#### 4.40.3.2 elapsed\_ticks

```
uint32_t elapsed_ticks = 0
```

#### 4.40.3.3 freqRamp\_left

```
rampa_struct freqRamp_left
```

**Initial value:**

```
= {  
    .in = 5.0F,  
    .Incr = TS*1000,  
}
```

#### 4.40.3.4 start\_ticks

```
uint32_t start_ticks = 0
```



- CONTROL.c, 99
- CONTROL.h, 30
- calculate\_derated\_current
  - REFERENCE.c, 148
  - REFERENCE.h, 81
- calibrate\_offsets
  - MEASUREMENTS.c, 123
  - MEASUREMENTS.h, 60
- CAN1\_RX0\_IRQHandler
  - stm32f7xx\_it.c, 154
- CAN\_e-Tech.c
  - handle\_CAN, 95
  - keepAlive, 97
  - send\_CAN\_message, 96
- CAN\_e-Tech.h
  - enableCAN, 27
  - handle\_CAN, 25
  - send\_CAN\_message, 26
- CANMessageInfo, 6
  - DLC, 6
  - getSig, 6
  - ID, 6
  - IDE, 7
- check\_motor\_parameters
  - MOTOR.c, 139
  - MOTOR.h, 68
- clear\_error
  - ERRORS.c, 102
  - ERRORS.h, 34
- constants
  - MotorParameters, 20
- CONTROL.c
  - calc\_current\_loop, 98
  - calc\_current\_reference, 99
  - calc\_duties, 99
  - saturate\_voltage, 100
- CONTROL.h
  - calc\_current\_loop, 29
  - calc\_current\_reference, 29
  - calc\_duties, 30
  - saturate\_voltage, 31
- CONTROL\_FAULT
  - ERRORS.h, 34
- cosTheta\_e
  - Encoder, 8
- CURRENT\_OFFSET
  - MEASUREMENTS.h, 60
- CURRENT\_SLOPE
  - MEASUREMENTS.h, 60
- currentOffsets
  - Analog, 5
- Da
  - Duties, 7
- DAC\_GPIO\_Port
  - main.h, 48
- DAC\_Pin
  - main.h, 48
- Db
  - Duties, 7
- Dc
  - Duties, 7
- DebugMon\_Handler
  - stm32f7xx\_it.c, 154
- derate\_current\_reference
  - REFERENCE.c, 148
  - REFERENCE.h, 82
- DIR\_GPIO\_Port
  - main.h, 48
- DIR\_Pin
  - main.h, 48
- DIR\_STATE
  - PCB\_IO.h, 72
- direction
  - InverterStruct, 12
- directionMeas
  - Encoder, 9
- DISABLE
  - PCB\_IO.h, 72
- disable\_control\_loops
  - INVERTER.c, 107
  - INVERTER.h, 41
- disable\_PWM
  - PWM.c, 145
  - PWM.h, 77
- DLC
  - CANMessageInfo, 6
- DMA2\_Stream0\_IRQHandler
  - stm32f7xx\_it.c, 155
- DMA2\_Stream1\_IRQHandler
  - stm32f7xx\_it.c, 155
- DMA2\_Stream2\_IRQHandler
  - stm32f7xx\_it.c, 155
- DT
  - INVERTER.h, 40
- dTorqueMax
  - MotorParameters, 20
- Duties, 7
  - Da, 7
  - Db, 7
  - Dc, 7
- duties
  - InverterStruct, 12
- eightTimesOneMinusXiSquared
  - MotorConstants, 16
- elapsed\_ticks
  - TASKS\_CRITICAL.c, 164
- ENABLE
  - PCB\_IO.h, 72
- enable
  - InverterStruct, 12
- enable\_control\_loops
  - INVERTER.c, 107
  - INVERTER.h, 41
- enable\_inverters
  - PCB\_IO.c, 142
  - PCB\_IO.h, 73

- ENABLE\_L\_GPIO\_Port
  - main.h, [48](#)
- ENABLE\_L\_Pin
  - main.h, [48](#)
- enable\_pin
  - InverterStruct, [12](#)
- enable\_port
  - InverterStruct, [12](#)
- enable\_PWM
  - PWM.c, [145](#)
  - PWM.h, [78](#)
- ENABLE\_R\_GPIO\_Port
  - main.h, [49](#)
- ENABLE\_R\_Pin
  - main.h, [49](#)
- enableCAN
  - CAN\_e-Tech.h, [27](#)
- enableSW
  - InverterStruct, [12](#)
- Encoder, [8](#)
  - A, [8](#)
  - B, [8](#)
  - cosTheta\_e, [8](#)
  - directionMeas, [9](#)
  - sinTheta\_e, [9](#)
  - theta\_e, [9](#)
  - we, [9](#)
  - Z, [9](#)
- encoder
  - InverterStruct, [12](#)
- Error\_Handler
  - main.c, [110](#)
  - main.h, [55](#)
- errors
  - InverterStruct, [13](#)
- ERRORS.c
  - clear\_error, [102](#)
  - is\_error\_set, [103](#)
  - set\_error, [103](#)
- ERRORS.h
  - clear\_error, [34](#)
  - CONTROL\_FAULT, [34](#)
  - FEEDBACK\_FAULT, [34](#)
  - InverterError, [33](#)
  - is\_error\_set, [34](#)
  - NONE, [34](#)
  - OVERCURRENT, [34](#)
  - OVERCURRENT\_TH, [33](#)
  - OVERSPEED, [34](#)
  - OVERSPEED\_TH, [33](#)
  - OVERTEMPERATURE\_INV, [34](#)
  - OVERTEMPERATURE\_INVERTER\_TH, [33](#)
  - OVERTEMPERATURE\_MOT, [34](#)
  - OVERTEMPERATURE\_MOTOR\_TH, [33](#)
  - OVERVOLTAGE, [34](#)
  - OVERVOLTAGE\_TH, [33](#)
  - POWER\_FAULT, [34](#)
  - set\_error, [35](#)
- UNDERVOLTAGE, [34](#)
- UNDERVOLTAGE\_TH, [33](#)
- WARNING, [34](#)
- eval\_inv\_FSM
  - FSM.c, [105](#)
  - FSM.h, [37](#)
- Feedback, [9](#)
  - idMeas, [10](#)
  - iqMeas, [10](#)
  - speedMeas, [10](#)
  - torqueCalc, [10](#)
- feedback
  - InverterStruct, [13](#)
- FEEDBACK\_FAULT
  - ERRORS.h, [34](#)
- fourTimesOneMinusXi
  - MotorConstants, [16](#)
- fourTimesOneMinusXiOnePlusXiSquared
  - MotorConstants, [17](#)
- fourTimesOneMinusXiXiSquared
  - MotorConstants, [17](#)
- freqRamp\_left
  - TASKS\_CRITICAL.c, [164](#)
- FSM.c
  - eval\_inv\_FSM, [105](#)
- FSM.h
  - eval\_inv\_FSM, [37](#)
- get\_currents\_voltage
  - MEASUREMENTS.c, [123](#)
  - MEASUREMENTS.h, [61](#)
- get\_idiq
  - MEASUREMENTS.c, [124](#)
  - MEASUREMENTS.h, [62](#)
- get\_linear
  - MEASUREMENTS.c, [125](#)
  - MEASUREMENTS.h, [63](#)
- get\_temperature
  - MEASUREMENTS.c, [125](#)
  - MEASUREMENTS.h, [64](#)
- getSig
  - CANMessageInfo, [6](#)
- hadc
  - InverterStruct, [13](#)
- handle\_CAN
  - CAN\_e-Tech.c, [95](#)
  - CAN\_e-Tech.h, [25](#)
- handle\_direction
  - PCB\_IO.c, [142](#)
  - PCB\_IO.h, [74](#)
- handle\_LED
  - PCB\_IO.c, [143](#)
  - PCB\_IO.h, [74](#)
- handle\_overtemperature\_faults
  - TASKS\_1ms.c, [159](#)
  - TASKS\_1ms.h, [88](#)
- handle\_torqueRef

- REFERENCE.c, [149](#)
- REFERENCE.h, [83](#)
- HardFault\_Handler
  - stm32f7xx\_it.c, [155](#)
- hcan1
  - stm32f7xx\_it.c, [157](#)
- hdac
  - stm32f7xx\_it.c, [157](#)
- hdma\_adc1
  - stm32f7xx\_it.c, [157](#)
- hdma\_adc2
  - stm32f7xx\_it.c, [157](#)
- hdma\_adc3
  - stm32f7xx\_it.c, [158](#)
- htim
  - InverterStruct, [13](#)
- htim1
  - stm32f7xx\_it.c, [158](#)
- htim6
  - stm32f7xx\_it.c, [158](#)
- ia
  - Analog, [5](#)
- ia\_L\_GPIO\_Port
  - main.h, [49](#)
- ia\_L\_Pin
  - main.h, [49](#)
- ia\_R\_GPIO\_Port
  - main.h, [49](#)
- ia\_R\_Pin
  - main.h, [49](#)
- ib
  - Analog, [5](#)
- ib\_L\_GPIO\_Port
  - main.h, [49](#)
- ib\_L\_Pin
  - main.h, [49](#)
- ib\_R\_GPIO\_Port
  - main.h, [49](#)
- ib\_R\_Pin
  - main.h, [49](#)
- ic
  - Analog, [6](#)
- ic\_L\_GPIO\_Port
  - main.h, [50](#)
- ic\_L\_Pin
  - main.h, [50](#)
- ic\_R\_GPIO\_Port
  - main.h, [50](#)
- ic\_R\_Pin
  - main.h, [50](#)
- ID
  - CANMessageInfo, [6](#)
- IDE
  - CANMessageInfo, [7](#)
- idLoop
  - InverterStruct, [13](#)
- idMeas
  - Feedback, [10](#)
- idRef
  - Reference, [22](#)
- iMax
  - MotorParameters, [20](#)
- init\_control\_loops
  - INVERTER.c, [107](#)
  - INVERTER.h, [41](#)
- initialize\_inverter
  - INVERTER.c, [107](#)
  - INVERTER.h, [42](#)
- INV\_STATE\_FAULT
  - INVERTER.h, [41](#)
- INV\_STATE\_IDLE
  - INVERTER.h, [41](#)
- INV\_STATE\_RUNNING
  - INVERTER.h, [41](#)
- INV\_STATE\_STARTUP
  - INVERTER.h, [41](#)
- INVERTER.c
  - disable\_control\_loops, [107](#)
  - enable\_control\_loops, [107](#)
  - init\_control\_loops, [107](#)
  - initialize\_inverter, [107](#)
  - inverter\_left, [109](#)
  - inverter\_right, [109](#)
- INVERTER.h
  - disable\_control\_loops, [41](#)
  - DT, [40](#)
  - enable\_control\_loops, [41](#)
  - init\_control\_loops, [41](#)
  - initialize\_inverter, [42](#)
  - INV\_STATE\_FAULT, [41](#)
  - INV\_STATE\_IDLE, [41](#)
  - INV\_STATE\_RUNNING, [41](#)
  - INV\_STATE\_STARTUP, [41](#)
  - inverter\_left, [43](#)
  - inverter\_right, [43](#)
  - InverterState, [40](#)
  - TS, [40](#)
- inverter\_left
  - INVERTER.c, [109](#)
  - INVERTER.h, [43](#)
- inverter\_right
  - INVERTER.c, [109](#)
  - INVERTER.h, [43](#)
- InverterError
  - ERRORS.h, [33](#)
- InverterState
  - INVERTER.h, [40](#)
- InverterStruct, [11](#)
  - analog, [12](#)
  - direction, [12](#)
  - duties, [12](#)
  - enable, [12](#)
  - enable\_pin, [12](#)
  - enable\_port, [12](#)
  - enableSW, [12](#)
  - encoder, [12](#)

- errors, [13](#)
- feedback, [13](#)
- hadc, [13](#)
- htim, [13](#)
- idLoop, [13](#)
- iqLoop, [13](#)
- led, [13](#)
- motor, [13](#)
- reference, [14](#)
- speedLoop, [14](#)
- state, [14](#)
- tempInverter, [14](#)
- tempMotor, [14](#)
- vd, [14](#)
- vq, [14](#)
- vsMax, [14](#)
- invThreePpLambda
  - MotorConstants, [17](#)
- invTorqueBase
  - MotorConstants, [17](#)
- iqLoop
  - InverterStruct, [13](#)
- iqMeas
  - Feedback, [10](#)
- iqRef
  - Reference, [22](#)
- is\_error\_set
  - ERRORS.c, [103](#)
  - ERRORS.h, [34](#)
- isc
  - MotorConstants, [17](#)
- isMaxRef
  - Reference, [22](#)
- J
  - MotorParameters, [20](#)
- keepAlive
  - CAN\_e-Tech.c, [97](#)
- lambda
  - MotorParameters, [20](#)
- lambdaDivLqMinusLd
  - MotorConstants, [17](#)
- Ld
  - MotorParameters, [20](#)
- LED, [15](#)
  - mode, [15](#)
  - pin, [15](#)
  - port, [15](#)
- led
  - InverterStruct, [13](#)
- LED\_ERR\_GPIO\_Port
  - main.h, [50](#)
- LED\_ERR\_Pin
  - main.h, [50](#)
- led\_left
  - PCB\_IO.c, [144](#)
  - PCB\_IO.h, [75](#)
- LED\_LEFT\_GPIO\_Port
  - main.h, [50](#)
- LED\_LEFT\_Pin
  - main.h, [50](#)
- LED\_MODE\_BLINK\_FAST
  - PCB\_IO.h, [73](#)
- LED\_MODE\_BLINK\_SLOW
  - PCB\_IO.h, [73](#)
- LED\_MODE\_OFF
  - PCB\_IO.h, [73](#)
- LED\_MODE\_ON
  - PCB\_IO.h, [73](#)
- led\_right
  - PCB\_IO.c, [144](#)
  - PCB\_IO.h, [75](#)
- LED\_RIGHT\_GPIO\_Port
  - main.h, [50](#)
- LED\_RIGHT\_Pin
  - main.h, [50](#)
- ledError
  - PCB\_IO.c, [144](#)
  - PCB\_IO.h, [75](#)
- LEDMode
  - PCB\_IO.h, [73](#)
- limit\_torque\_to\_prevent\_overspeed
  - REFERENCE.c, [150](#)
  - REFERENCE.h, [84](#)
- Lq
  - MotorParameters, [21](#)
- main
  - main.c, [111](#)
- main.c
  - Error\_Handler, [110](#)
  - main, [111](#)
  - SystemClock\_Config, [111](#)
- main.h
  - A\_L\_GPIO\_Port, [47](#)
  - A\_L\_Pin, [47](#)
  - A\_R\_GPIO\_Port, [47](#)
  - A\_R\_Pin, [47](#)
  - B\_L\_GPIO\_Port, [48](#)
  - B\_L\_Pin, [48](#)
  - B\_R\_GPIO\_Port, [48](#)
  - B\_R\_Pin, [48](#)
  - DAC\_GPIO\_Port, [48](#)
  - DAC\_Pin, [48](#)
  - DIR\_GPIO\_Port, [48](#)
  - DIR\_Pin, [48](#)
  - ENABLE\_L\_GPIO\_Port, [48](#)
  - ENABLE\_L\_Pin, [48](#)
  - ENABLE\_R\_GPIO\_Port, [49](#)
  - ENABLE\_R\_Pin, [49](#)
  - Error\_Handler, [55](#)
  - ia\_L\_GPIO\_Port, [49](#)
  - ia\_L\_Pin, [49](#)
  - ia\_R\_GPIO\_Port, [49](#)
  - ia\_R\_Pin, [49](#)
  - ib\_L\_GPIO\_Port, [49](#)

- ib\_L\_Pin, [49](#)
- ib\_R\_GPIO\_Port, [49](#)
- ib\_R\_Pin, [49](#)
- ic\_L\_GPIO\_Port, [50](#)
- ic\_L\_Pin, [50](#)
- ic\_R\_GPIO\_Port, [50](#)
- ic\_R\_Pin, [50](#)
- LED\_ERR\_GPIO\_Port, [50](#)
- LED\_ERR\_Pin, [50](#)
- LED\_LEFT\_GPIO\_Port, [50](#)
- LED\_LEFT\_Pin, [50](#)
- LED\_RIGHT\_GPIO\_Port, [50](#)
- LED\_RIGHT\_Pin, [50](#)
- PWM1\_L\_GPIO\_Port, [51](#)
- PWM1\_L\_Pin, [51](#)
- PWM1\_R\_GPIO\_Port, [51](#)
- PWM1\_R\_Pin, [51](#)
- PWM2\_L\_GPIO\_Port, [51](#)
- PWM2\_L\_Pin, [51](#)
- PWM2\_R\_GPIO\_Port, [51](#)
- PWM2\_R\_Pin, [51](#)
- PWM3\_L\_GPIO\_Port, [51](#)
- PWM3\_L\_Pin, [51](#)
- PWM3\_R\_GPIO\_Port, [52](#)
- PWM3\_R\_Pin, [52](#)
- PWM4\_L\_GPIO\_Port, [52](#)
- PWM4\_L\_Pin, [52](#)
- PWM4\_R\_GPIO\_Port, [52](#)
- PWM4\_R\_Pin, [52](#)
- PWM5\_L\_GPIO\_Port, [52](#)
- PWM5\_L\_Pin, [52](#)
- PWM5\_R\_GPIO\_Port, [52](#)
- PWM5\_R\_Pin, [52](#)
- PWM6\_L\_GPIO\_Port, [53](#)
- PWM6\_L\_Pin, [53](#)
- PWM6\_R\_GPIO\_Port, [53](#)
- PWM6\_R\_Pin, [53](#)
- SC\_det\_GPIO\_Port, [53](#)
- SC\_det\_Pin, [53](#)
- Tinv\_L\_GPIO\_Port, [53](#)
- Tinv\_L\_Pin, [53](#)
- Tinv\_R\_GPIO\_Port, [53](#)
- Tinv\_R\_Pin, [53](#)
- Tmot\_L\_GPIO\_Port, [54](#)
- Tmot\_L\_Pin, [54](#)
- Tmot\_R\_GPIO\_Port, [54](#)
- Tmot\_R\_Pin, [54](#)
- TRIP\_L\_GPIO\_Port, [54](#)
- TRIP\_L\_Pin, [54](#)
- TRIP\_R\_GPIO\_Port, [54](#)
- TRIP\_R\_Pin, [54](#)
- VDC\_L\_GPIO\_Port, [54](#)
- VDC\_L\_Pin, [54](#)
- VDC\_R\_GPIO\_Port, [55](#)
- VDC\_R\_Pin, [55](#)
- WRN\_L\_GPIO\_Port, [55](#)
- WRN\_L\_Pin, [55](#)
- WRN\_R\_GPIO\_Port, [55](#)
- WRN\_R\_Pin, [55](#)
- Z\_L\_GPIO\_Port, [55](#)
- Z\_L\_Pin, [55](#)
- Z\_R\_GPIO\_Port, [55](#)
- Z\_R\_Pin, [55](#)
- MEASUREMENTS.c
  - calibrate\_offsets, [123](#)
  - get\_currents\_voltage, [123](#)
  - get\_idiq, [124](#)
  - get\_linear, [125](#)
  - get\_temperature, [125](#)
  - rawADC\_left, [126](#)
  - rawADC\_right, [126](#)
  - rawADC\_temp, [126](#)
  - templInverterLUT, [127](#)
  - tempMotorLUT, [132](#)
- MEASUREMENTS.h
  - calibrate\_offsets, [60](#)
  - CURRENT\_OFFSET, [60](#)
  - CURRENT\_SLOPE, [60](#)
  - get\_currents\_voltage, [61](#)
  - get\_idiq, [62](#)
  - get\_linear, [63](#)
  - get\_temperature, [64](#)
  - rawADC\_left, [64](#)
  - rawADC\_right, [64](#)
  - rawADC\_temp, [65](#)
  - templInverterLUT, [65](#)
  - tempMotorLUT, [65](#)
  - VOLTAGE\_OFFSET, [60](#)
  - VOLTAGE\_SLOPE, [60](#)
- MemManage\_Handler
  - stm32f7xx\_it.c, [156](#)
- mode
  - LED, [15](#)
- motor
  - InverterStruct, [13](#)
- MOTOR.c
  - check\_motor\_parameters, [139](#)
  - motor\_left, [140](#)
  - motor\_right, [140](#)
  - precalculate\_motor\_constants, [139](#)
- MOTOR.h
  - check\_motor\_parameters, [68](#)
  - motor\_left, [69](#)
  - motor\_right, [69](#)
  - precalculate\_motor\_constants, [69](#)
- motor\_left
  - MOTOR.c, [140](#)
  - MOTOR.h, [69](#)
- motor\_right
  - MOTOR.c, [140](#)
  - MOTOR.h, [69](#)
- MotorConstants, [16](#)
  - betaMinusIsc, [16](#)
  - eightTimesOneMinusXiSquared, [16](#)
  - fourTimesOneMinusXi, [16](#)
  - fourTimesOneMinusXiOnePlusXiSquared, [17](#)



- fourTimesOneMinusXiXiSquared, [17](#)
- invThreePpLambda, [17](#)
- invTorqueBase, [17](#)
- isc, [17](#)
- lambdaDivLqMinusLd, [17](#)
- oneMinusXi, [17](#)
- threePpLambda, [17](#)
- threePpLdMinusLq, [18](#)
- torqueBase, [18](#)
- twoMinusXi, [18](#)
- twoMinusXiSquared, [18](#)
- twoTimesOneMinusXiOnePlusXiSquared, [18](#)
- twoTimesOneMinusXiXiSquared, [18](#)
- xi, [18](#)
- xiSquared, [18](#)
- MotorParameters, [19](#)
  - b, [20](#)
  - constants, [20](#)
  - dTorqueMax, [20](#)
  - iMax, [20](#)
  - J, [20](#)
  - lambda, [20](#)
  - Ld, [20](#)
  - Lq, [21](#)
  - pp, [21](#)
  - Rs, [21](#)
  - speedMax\_RPM, [21](#)
  - torqueMax, [21](#)
  - vDCMax, [21](#)
- NMI\_Handler
  - stm32f7xx\_it.c, [156](#)
- NONE
  - ERRORS.h, [34](#)
- oneMinusXi
  - MotorConstants, [17](#)
- OVERCURRENT
  - ERRORS.h, [34](#)
- OVERCURRENT\_TH
  - ERRORS.h, [33](#)
- OVERSPEED
  - ERRORS.h, [34](#)
- OVERSPEED\_TH
  - ERRORS.h, [33](#)
- OVERTEMPERATURE\_INV
  - ERRORS.h, [34](#)
- OVERTEMPERATURE\_INVERTER\_TH
  - ERRORS.h, [33](#)
- OVERTEMPERATURE\_MOT
  - ERRORS.h, [34](#)
- OVERTEMPERATURE\_MOTOR\_TH
  - ERRORS.h, [33](#)
- OVERVOLTAGE
  - ERRORS.h, [34](#)
- OVERVOLTAGE\_TH
  - ERRORS.h, [33](#)
- PCB\_IO.c
  - enable\_inverters, [142](#)
  - handle\_direction, [142](#)
  - handle\_LED, [143](#)
  - led\_left, [144](#)
  - led\_right, [144](#)
  - ledError, [144](#)
- PCB\_IO.h
  - DIR\_STATE, [72](#)
  - DISABLE, [72](#)
  - ENABLE, [72](#)
  - enable\_inverters, [73](#)
  - handle\_direction, [74](#)
  - handle\_LED, [74](#)
  - led\_left, [75](#)
  - LED\_MODE\_BLINK\_FAST, [73](#)
  - LED\_MODE\_BLINK\_SLOW, [73](#)
  - LED\_MODE\_OFF, [73](#)
  - LED\_MODE\_ON, [73](#)
  - led\_right, [75](#)
  - ledError, [75](#)
  - LEDMode, [73](#)
  - SC\_DET\_STATE, [72](#)
  - WRN\_STATE, [73](#)
- PendSV\_Handler
  - stm32f7xx\_it.c, [156](#)
- pin
  - LED, [15](#)
- port
  - LED, [15](#)
- POWER\_FAULT
  - ERRORS.h, [34](#)
- pp
  - MotorParameters, [21](#)
- precalculate\_motor\_constants
  - MOTOR.c, [139](#)
  - MOTOR.h, [69](#)
- PWM.c
  - disable\_PWM, [145](#)
  - enable\_PWM, [145](#)
  - update\_PWM, [145](#)
- PWM.h
  - disable\_PWM, [77](#)
  - enable\_PWM, [78](#)
  - update\_PWM, [78](#)
- PWM1\_L\_GPIO\_Port
  - main.h, [51](#)
- PWM1\_L\_Pin
  - main.h, [51](#)
- PWM1\_R\_GPIO\_Port
  - main.h, [51](#)
- PWM1\_R\_Pin
  - main.h, [51](#)
- PWM2\_L\_GPIO\_Port
  - main.h, [51](#)
- PWM2\_L\_Pin
  - main.h, [51](#)
- PWM2\_R\_GPIO\_Port
  - main.h, [51](#)

- PWM2\_R\_Pin
  - main.h, [51](#)
- PWM3\_L\_GPIO\_Port
  - main.h, [51](#)
- PWM3\_L\_Pin
  - main.h, [51](#)
- PWM3\_R\_GPIO\_Port
  - main.h, [52](#)
- PWM3\_R\_Pin
  - main.h, [52](#)
- PWM4\_L\_GPIO\_Port
  - main.h, [52](#)
- PWM4\_L\_Pin
  - main.h, [52](#)
- PWM4\_R\_GPIO\_Port
  - main.h, [52](#)
- PWM4\_R\_Pin
  - main.h, [52](#)
- PWM5\_L\_GPIO\_Port
  - main.h, [52](#)
- PWM5\_L\_Pin
  - main.h, [52](#)
- PWM5\_R\_GPIO\_Port
  - main.h, [52](#)
- PWM5\_R\_Pin
  - main.h, [52](#)
- PWM6\_L\_GPIO\_Port
  - main.h, [53](#)
- PWM6\_L\_Pin
  - main.h, [53](#)
- PWM6\_R\_GPIO\_Port
  - main.h, [53](#)
- PWM6\_R\_Pin
  - main.h, [53](#)
- rawADC\_left
  - MEASUREMENTS.c, [126](#)
  - MEASUREMENTS.h, [64](#)
- rawADC\_right
  - MEASUREMENTS.c, [126](#)
  - MEASUREMENTS.h, [64](#)
- rawADC\_temp
  - MEASUREMENTS.c, [126](#)
  - MEASUREMENTS.h, [65](#)
- read\_temperatures
  - TASKS\_1ms.c, [160](#)
  - TASKS\_1ms.h, [89](#)
- Reference, [22](#)
  - idRef, [22](#)
  - iqRef, [22](#)
  - isMaxRef, [22](#)
  - torqueRef, [22](#)
- reference
  - InverterStruct, [14](#)
- REFERENCE.c
  - calculate\_derated\_current, [148](#)
  - derate\_current\_reference, [148](#)
  - handle\_torqueRef, [149](#)
  - limit\_torque\_to\_prevent\_overspeed, [150](#)
  - saturate\_symmetric, [151](#)
  - set\_torque\_direction, [151](#)
  - torqueRefIn\_left, [152](#)
  - torqueRefIn\_right, [152](#)
- REFERENCE.h
  - calculate\_derated\_current, [81](#)
  - derate\_current\_reference, [82](#)
  - handle\_torqueRef, [83](#)
  - limit\_torque\_to\_prevent\_overspeed, [84](#)
  - saturate\_symmetric, [85](#)
  - set\_torque\_direction, [85](#)
  - TEMP\_INVERTER\_DERATING, [81](#)
  - TEMP\_INVERTER\_MAX, [81](#)
  - TEMP\_MOTOR\_DERATING, [81](#)
  - TEMP\_MOTOR\_MAX, [81](#)
  - torqueRefIn\_left, [86](#)
  - torqueRefIn\_right, [86](#)
- Rs
  - MotorParameters, [21](#)
- saturate\_symmetric
  - REFERENCE.c, [81](#)
  - REFERENCE.h, [85](#)
- saturate\_voltage
  - CONTROL.c, [100](#)
  - CONTROL.h, [31](#)
- SC\_det\_GPIO\_Port
  - main.h, [53](#)
- SC\_det\_Pin
  - main.h, [53](#)
- SC\_DET\_STATE
  - PCB\_IO.h, [72](#)
- send\_CAN\_message
  - CAN\_e-Tech.c, [96](#)
  - CAN\_e-Tech.h, [26](#)
- set\_error
  - ERRORS.c, [103](#)
  - ERRORS.h, [35](#)
- set\_torque\_direction
  - REFERENCE.c, [151](#)
  - REFERENCE.h, [85](#)
- sinTheta\_e
  - Encoder, [9](#)
- speedLoop
  - InverterStruct, [14](#)
- speedMax\_RPM
  - MotorParameters, [21](#)
- speedMeas
  - Feedback, [10](#)
- start\_ticks
  - TASKS\_CRITICAL.c, [164](#)
- state
  - InverterStruct, [14](#)
- stm32f7xx\_it.c
  - BusFault\_Handler, [154](#)
  - CAN1\_RX0\_IRQHandler, [154](#)
  - DebugMon\_Handler, [154](#)
  - DMA2\_Stream0\_IRQHandler, [155](#)
  - DMA2\_Stream1\_IRQHandler, [155](#)

- DMA2\_Stream2\_IRQHandler, [155](#)
- HardFault\_Handler, [155](#)
- hcan1, [157](#)
- hdac, [157](#)
- hdma\_adc1, [157](#)
- hdma\_adc2, [157](#)
- hdma\_adc3, [158](#)
- htim1, [158](#)
- htim6, [158](#)
- MemManage\_Handler, [156](#)
- NMI\_Handler, [156](#)
- PendSV\_Handler, [156](#)
- SVC\_Handler, [156](#)
- SysTick\_Handler, [156](#)
- TIM1\_UP\_TIM10\_IRQHandler, [156](#)
- TIM6\_DAC\_IRQHandler, [156](#)
- UsageFault\_Handler, [157](#)
- SVC\_Handler
  - stm32f7xx\_it.c, [156](#)
- SystemClock\_Config
  - main.c, [111](#)
- SysTick\_Handler
  - stm32f7xx\_it.c, [156](#)
- tasks\_1ms
  - TASKS\_1ms.c, [160](#)
  - TASKS\_1ms.h, [90](#)
- TASKS\_1ms.c
  - handle\_overtemperature\_faults, [159](#)
  - read\_temperatures, [160](#)
  - tasks\_1ms, [160](#)
- TASKS\_1ms.h
  - handle\_overtemperature\_faults, [88](#)
  - read\_temperatures, [89](#)
  - tasks\_1ms, [90](#)
- TASKS\_CRITICAL.c
  - angle\_left, [164](#)
  - elapsed\_ticks, [164](#)
  - freqRamp\_left, [164](#)
  - start\_ticks, [164](#)
  - tasks\_critical\_left, [163](#)
  - tasks\_critical\_right, [163](#)
- TASKS\_CRITICAL.h
  - tasks\_critical\_left, [92](#)
  - tasks\_critical\_right, [92](#)
- tasks\_critical\_left
  - TASKS\_CRITICAL.c, [163](#)
  - TASKS\_CRITICAL.h, [92](#)
- tasks\_critical\_right
  - TASKS\_CRITICAL.c, [163](#)
  - TASKS\_CRITICAL.h, [92](#)
- TEMP\_INVERTER\_DERATING
  - REFERENCE.h, [81](#)
- TEMP\_INVERTER\_MAX
  - REFERENCE.h, [81](#)
- TEMP\_MOTOR\_DERATING
  - REFERENCE.h, [81](#)
- TEMP\_MOTOR\_MAX
  - REFERENCE.h, [81](#)
- tempInverter
  - InverterStruct, [14](#)
- tempInverterLUT
  - MEASUREMENTS.c, [127](#)
  - MEASUREMENTS.h, [65](#)
- tempMotor
  - InverterStruct, [14](#)
- tempMotorLUT
  - MEASUREMENTS.c, [132](#)
  - MEASUREMENTS.h, [65](#)
- theta\_e
  - Encoder, [9](#)
- threePpLambda
  - MotorConstants, [17](#)
- threePpLdMinusLq
  - MotorConstants, [18](#)
- TIM1\_UP\_TIM10\_IRQHandler
  - stm32f7xx\_it.c, [156](#)
- TIM6\_DAC\_IRQHandler
  - stm32f7xx\_it.c, [156](#)
- Tinv\_L\_GPIO\_Port
  - main.h, [53](#)
- Tinv\_L\_Pin
  - main.h, [53](#)
- Tinv\_R\_GPIO\_Port
  - main.h, [53](#)
- Tinv\_R\_Pin
  - main.h, [53](#)
- Tmot\_L\_GPIO\_Port
  - main.h, [54](#)
- Tmot\_L\_Pin
  - main.h, [54](#)
- Tmot\_R\_GPIO\_Port
  - main.h, [54](#)
- Tmot\_R\_Pin
  - main.h, [54](#)
- torqueBase
  - MotorConstants, [18](#)
- torqueCalc
  - Feedback, [10](#)
- torqueMax
  - MotorParameters, [21](#)
- torqueRef
  - Reference, [22](#)
- torqueRefIn\_left
  - REFERENCE.c, [152](#)
  - REFERENCE.h, [86](#)
- torqueRefIn\_right
  - REFERENCE.c, [152](#)
  - REFERENCE.h, [86](#)
- TRIP\_L\_GPIO\_Port
  - main.h, [54](#)
- TRIP\_L\_Pin
  - main.h, [54](#)
- TRIP\_R\_GPIO\_Port
  - main.h, [54](#)
- TRIP\_R\_Pin
  - main.h, [54](#)

TS  
     INVERTER.h, 40  
 twoMinusXi  
     MotorConstants, 18  
 twoMinusXiSquared  
     MotorConstants, 18  
 twoTimesOneMinusXiOnePlusXiSquared  
     MotorConstants, 18  
 twoTimesOneMinusXiXiSquared  
     MotorConstants, 18  
  
 UNDERVOLTAGE  
     ERRORS.h, 34  
 UNDERVOLTAGE\_TH  
     ERRORS.h, 33  
 update\_PWM  
     PWM.c, 145  
     PWM.h, 78  
 UsageFault\_Handler  
     stm32f7xx\_it.c, 157  
  
 vd  
     InverterStruct, 14  
 vDC  
     Analog, 6  
 VDC\_L\_GPIO\_Port  
     main.h, 54  
 VDC\_L\_Pin  
     main.h, 54  
 VDC\_R\_GPIO\_Port  
     main.h, 55  
 VDC\_R\_Pin  
     main.h, 55  
 vDCMax  
     MotorParameters, 21  
 VOLTAGE\_OFFSET  
     MEASUREMENTS.h, 60  
 VOLTAGE\_SLOPE  
     MEASUREMENTS.h, 60  
 vq  
     InverterStruct, 14  
 vsMax  
     InverterStruct, 14  
  
 WARNING  
     ERRORS.h, 34  
 we  
     Encoder, 9  
 WRN\_L\_GPIO\_Port  
     main.h, 55  
 WRN\_L\_Pin  
     main.h, 55  
 WRN\_R\_GPIO\_Port  
     main.h, 55  
 WRN\_R\_Pin  
     main.h, 55  
 WRN\_STATE  
     PCB\_IO.h, 73  
  
 xi  
     MotorConstants, 18  
 xiSquared  
     MotorConstants, 18  
  
 Z  
     Encoder, 9  
 Z\_L\_GPIO\_Port  
     main.h, 55  
 Z\_L\_Pin  
     main.h, 55  
 Z\_R\_GPIO\_Port  
     main.h, 55  
 Z\_R\_Pin  
     main.h, 55