e-Tech Racing's Inverter Firmware

v0

Generated by Doxygen 1.10.0

Chapter 1

Topic Index

1.1 Topics

Here is a list of all topics with brief descriptions:

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

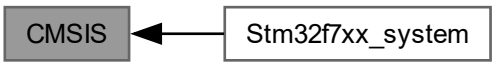
Here is a list of all files with brief descriptions:

Chapter 4

Topic Documentation

4.1 CMSIS

Collaboration diagram for CMSIS:



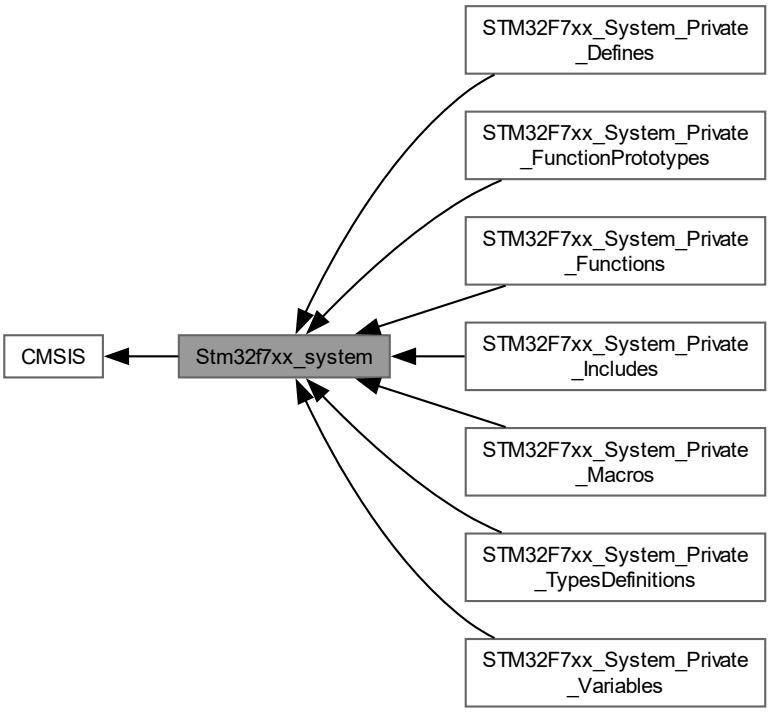
Topics

- Stm32f7xx_system

4.1.1 Detailed Description

4.1.2 Stm32f7xx_system

Collaboration diagram for Stm32f7xx_system:



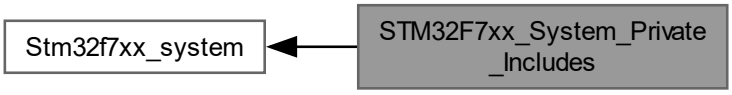
Topics

- STM32F7xx_System_Private_Includes
- STM32F7xx_System_Private_TypesDefinitions
- STM32F7xx_System_Private_Defines
- STM32F7xx_System_Private_Macros
- STM32F7xx_System_Private_Variables
- STM32F7xx_System_Private_FunctionPrototypes
- STM32F7xx_System_Private_Functions

4.1.2.1 Detailed Description

4.1.2.2 STM32F7xx_System_Private_Includes

Collaboration diagram for STM32F7xx_System_Private_Includes:



Macros

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

4.1.2.2.1 Detailed Description

4.1.2.2.2 Macro Definition Documentation

4.1.2.2.2.1 HSE_VALUE

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

Default value of the External oscillator in Hz

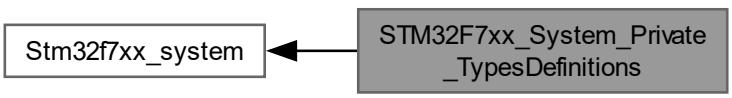
4.1.2.2.2.2 HSI_VALUE

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

Value of the Internal oscillator in Hz

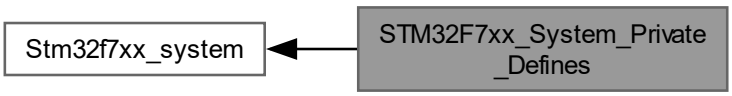
4.1.2.3 STM32F7xx_System_Private_TypesDefinitions

Collaboration diagram for STM32F7xx_System_Private_TypesDefinitions:



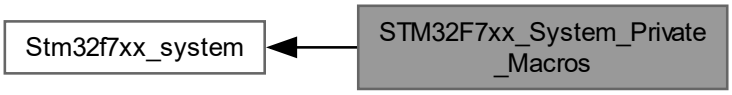
4.1.2.4 STM32F7xx_System_Private_Defines

Collaboration diagram for STM32F7xx_System_Private_Defines:



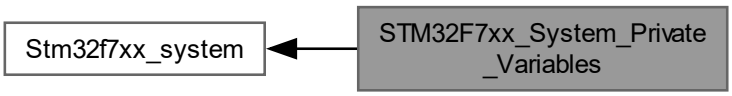
4.1.2.5 STM32F7xx_System_Private_Macros

Collaboration diagram for STM32F7xx_System_Private_Macros:



4.1.2.6 STM32F7xx_System_Private_Variables

Collaboration diagram for STM32F7xx_System_Private_Variables:



Variables

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

4.1.2.6.1 Detailed Description

4.1.2.6.2 Variable Documentation

4.1.2.6.2.1 AHBPrescTable

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

4.1.2.6.2.2 APBPrescTable

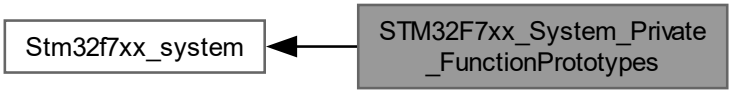
```
const uint8_t APBPrescTable[8] = {0, 0, 0, 0, 1, 2, 3, 4}
```

4.1.2.6.2.3 SystemCoreClock

```
uint32_t SystemCoreClock = 16000000
```

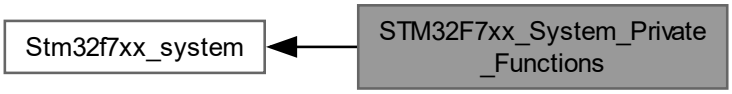
4.1.2.7 STM32F7xx_System_Private_FunctionPrototypes

Collaboration diagram for STM32F7xx_System_Private_FunctionPrototypes:



4.1.2.8 STM32F7xx_System_Private_Functions

Collaboration diagram for STM32F7xx_System_Private_Functions:



Functions

- void SystemInit (void)

  Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.
- void SystemCoreClockUpdate (void)

  Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

4.1.2.8.1 Detailed Description

4.1.2.8.2 Function Documentation

4.1.2.8.2.1 SystemCoreClockUpdate()

```
void SystemCoreClockUpdate (
            void  )
```

Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

Note

Each time the core clock (HCLK) changes, this function must be called to update SystemCoreClock variable value. Otherwise, any configuration based on this variable will be incorrect.

- The system frequency computed by this function is not the real frequency in the chip. It is calculated based on the predefined constant and the selected clock source:

- If SYSCLK source is HSI, SystemCoreClock will contain the HSI_VALUE(∗)
- If SYSCLK source is HSE, SystemCoreClock will contain the HSE_VALUE(∗∗)
- If SYSCLK source is PLL, SystemCoreClock will contain the HSE_VALUE(∗∗) or HSI_VALUE(∗) multiplied/divided by the PLL factors.

(∗) HSI_VALUE is a constant defined in stm32f7xx_hal_conf.h file (default value 16 MHz) but the real value may vary depending on the variations in voltage and temperature.

(∗∗) HSE_VALUE is a constant defined in stm32f7xx_hal_conf.h file (default value 25 MHz), user has to ensure that HSE_VALUE is same as the real frequency of the crystal used. Otherwise, this function may have wrong result.

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

Parameters

None	

Return values

None	

4.1.2.8.2.2 SystemInit()

```
void SystemInit (
             void  )
```

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

Parameters

None	

Return values

None	

Chapter 5

Data Structure Documentation

5.1 angle_struct Struct Reference

`#include <Pergamon_float.h>`

Data Fields

- float freq
- float Ts
- float angle
- void(∗ calc)()

5.1.1 Field Documentation

5.1.1.1 angle

`float angle`

5.1.1.2 calc

`void(* calc) ()`

5.1.1.3 freq

`float freq`

5.1.1.4 Ts

`float Ts`

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.2 avg_struct_10 Struct Reference

`#include <Pergamon_float.h>`

Data Fields

- float out
- float in [10]

5.2.1 Field Documentation

5.2.1.1 in

`float in[10]`

5.2.1.2 out

`float out`

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.3 clarke3F_struct Struct Reference

`#include <Pergamon_float.h>`

Data Fields

- float a
- float b
- float D
- float Q
- void(∗ calc)()

5.3.1 Field Documentation

5.3.1.1 a

`float a`

5.3.1.2 b

`float b`

5.3.1.3 calc

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

5.3.1.4 D

```
float D
```

5.3.1.5 Q

```
float Q
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.4 datalog_struct Struct Reference

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

Data Fields

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

5.4.1 Field Documentation

5.4.1.1 calc

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

5.4.1.2 estat

```
uint16_t estat
```

5.4.1.3 i

```
uint16_t i
```

5.4.1.4 j

```
uint16_t j
```

5.4.1.5 log

```
float log[N_DATALOG]
```

5.4.1.6 prescaler

```
uint16_t prescaler
```

5.4.1.7 var

```
float* var
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.5 Encoder Struct Reference

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

Data Fields

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

5.5.1 Field Documentation

5.5.1.1 A

```
uint16_t A
```

Encoder channel A value

5.5.1.2 B

```
uint16_t B
```

Encoder channel B value

5.5.1.3 DIR

```
uint8_t DIR
```

Direction

5.5.1.4 theta_e

```
float theta_e
```

Electrical rotor position

5.5.1.5 theta_m

```
float theta_m
```

Mechanical rotor position

5.5.1.6 we

```
float we
```

Electrical angular velocity

5.5.1.7 wm_rpm

```
float wm_rpm
```

Mechanical angular velocity (RPM)

5.5.1.8 Z

```
uint16_t Z
```

Encoder channel Z value

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/MEASUREMENTS.h

5.6 filtreLP_struct Struct Reference

`#include <Pergamon_float.h>`

Data Fields

- float in
- float out
- float alfa
- float Ts
- float fc
- uint16_t enable
- void(* init)()
- void(* calc)()

5.6.1 Field Documentation

5.6.1.1 alfa

```
float alfa
```

5.6.1.2 calc

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

5.6.1.3 enable

```
uint16_t enable
```

5.6.1.4 fc

```
float fc
```

5.6.1.5 in

```
float in
```

5.6.1.6 init

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

5.6.1.7 out

```
float out
```

5.6.1.8 Ts

```
float Ts
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.7 iclarke3F_struct Struct Reference

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

Data Fields

- float D
- float Q
- float a
- float b
- void(∗ calc)()

5.7.1 Field Documentation

5.7.1.1 a

```
float a
```

5.7.1.2 b

```
float b
```

5.7.1.3 calc

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

5.7.1.4 D

```
float D
```

5.7.1.5 Q

```
float Q
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.8 InverterOperation Struct Reference

Inverter operation structure.

`#include <FSM.h>`

Data Fields

- InverterOperationState state
- GPIO_TypeDef * LED_port
- uint16_t LED_pin
- GPIO_TypeDef * enable_port
- uint16_t enable_pin

5.8.1 Detailed Description

Inverter operation structure.

5.8.2 Field Documentation

5.8.2.1 enable_pin

`uint16_t enable_pin`

Pin number for enabling/disabling the inverter

5.8.2.2 enable_port

`GPIO_TypeDef* enable_port`

GPIO port for enabling/disabling the inverter

5.8.2.3 LED_pin

`uint16_t LED_pin`

Pin number for controlling the LED

5.8.2.4 LED_port

`GPIO_TypeDef* LED_port`

GPIO port for controlling the LED

5.8.2.5 state

```
InverterOperationState state
```

Current state of the inverter operation

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/FSM.h

5.9 irot_struct Struct Reference

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

Data Fields

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

5.9.1 Field Documentation

5.9.1.1 calc

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

5.9.1.2 cosFi

```
float cosFi
```

5.9.1.3 d

```
float d
```

5.9.1.4 D

```
float D
```

5.9.1.5 q

```
float q
```

5.9.1.6 Q

```
float Q
```

5.9.1.7 sinFi

```
float sinFi
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.10 LED Struct Reference

LED structure.

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

Data Fields

- GPIO_TypeDef * port
- uint16_t pin
- LEDMode mode

5.10.1 Detailed Description

LED structure.

5.10.2 Field Documentation

5.10.2.1 mode

```
LEDMode mode
```

Current LED mode

5.10.2.2 pin

```
uint16_t pin
```

Pin number for controlling the LED

5.10.2.3 port

```
GPIO_TypeDef* port
```

GPIO port for controlling the LED

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PCB_IO.h

5.11 Measurements Struct Reference

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

Data Fields

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

5.11.1 Field Documentation

5.11.1.1 ia

```
float ia
```

Phase A current

5.11.1.2 ib

```
float ib
```

Phase B current

5.11.1.3 ic

```
float ic
```

Phase C current

5.11.1.4 VDC

```
float VDC
```

DC link voltage

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/MEASUREMENTS.h

5.12 pi_aw_struct Struct Reference

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

Data Fields

- uint16_t enable
- float Ts
- float Kp
- float Ki
- float Kaw
- float e [2]
- float pi_consig
- float pi_fdb
- float pi_out_max
- float pi_out_min
- float pi_out_presat
- float pi_out_postsat
- float pi_out
- float pi_int [2]
- float pi_ffw [2]
- void(∗ calc)()

5.12.1 Field Documentation

5.12.1.1 calc

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

5.12.1.2 e

```
float e[2]
```

5.12.1.3 enable

```
uint16_t enable
```

5.12.1.4 Kaw

```
float Kaw
```

5.12.1.5 Ki

```
float Ki
```

5.12.1.6 Kp

```
float Kp
```

5.12.1.7 pi_consig

```
float pi_consig
```

5.12.1.8 pi_fdb

```
float pi_fdb
```

5.12.1.9 pi_ffw

```
float pi_ffw[2]
```

5.12.1.10 pi_int

```
float pi_int[2]
```

5.12.1.11 pi_out

```
float pi_out
```

5.12.1.12 pi_out_max

```
float pi_out_max
```

5.12.1.13 pi_out_min

```
float pi_out_min
```

5.12.1.14 pi_out_postsat

```
float pi_out_postsat
```

5.12.1.15 pi_out_presat

```
float pi_out_presat
```

5.12.1.16 Ts

```
float Ts
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.13 pi_struct Struct Reference

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

Data Fields

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

5.13.1 Field Documentation

5.13.1.1 calc

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

5.13.1.2 e

```
float e[2]
```

5.13.1.3 enable

```
uint16_t enable
```

5.13.1.4 init

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

5.13.1.5 K0

```
float K0
```

5.13.1.6 K1

```
float K1
```

5.13.1.7 Ki

```
float Ki
```

5.13.1.8 Kp

```
float Kp
```

5.13.1.9 pi_consig

```
float pi_consig
```

5.13.1.10 pi_fdb

```
float pi_fdb
```

5.13.1.11 pi_ffw

```
float pi_ffw[2]
```

5.13.1.12 pi_out

```
float pi_out
```

5.13.1.13 pi_out_max

```
float pi_out_max
```

5.13.1.14 pi_out_min

```
float pi_out_min
```

5.13.1.15 Ts

```
float Ts
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.14 rampa_dual_struct Struct Reference

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

Data Fields

- float in
- float out
- float Incr
- float Decr
- uint8_t enable
- void(∗ calc)()

5.14.1 Field Documentation

5.14.1.1 calc

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

5.14.1.2 Decr

```
float Decr
```

5.14.1.3 enable

```
uint8_t enable
```

5.14.1.4 in

```
float in
```

5.14.1.5 Incr

```
float Incr
```

5.14.1.6 out

```
float out
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.15 rampa_struct Struct Reference

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

Data Fields

- float in
- float out
- float Incr
- uint8_t enable
- void(* calc)()

5.15.1 Field Documentation

5.15.1.1 calc

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

5.15.1.2 enable

```
uint8_t enable
```

5.15.1.3 in

```
float in
```

5.15.1.4 Incr

```
float Incr
```

5.15.1.5 out

```
float out
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.16 RMS_struct Struct Reference

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

Data Fields

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

5.16.1 Field Documentation

5.16.1.1 Angle

```
float Angle
```

5.16.1.2 Angle_ant

```
float Angle_ant
```

5.16.1.3 Freq

```
float Freq
```

5.16.1.4 Measure

```
float Measure
```

5.16.1.5 Out_RMS

```
float Out_RMS
```

5.16.1.6 Sq_Sum

```
float Sq_Sum
```

5.16.1.7 T_exec

```
float T_exec
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.17 rot_struct Struct Reference

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

Data Fields

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

5.17.1 Field Documentation

5.17.1.1 calc

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

5.17.1.2 cosFi

```
float cosFi
```

5.17.1.3 D

```
float D
```

5.17.1.4 d

```
float d
```

5.17.1.5 Q

```
float Q
```

5.17.1.6 q

```
float q
```

5.17.1.7 sinFi

```
float sinFi
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.18 step_struct Struct Reference

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

Data Fields

- float fs
- float In
- float Out
- float Step
- float t_step
- uint32_t Pulses
- uint32_t Counter
- uint16_t enable
- void(* calc)()

5.18.1 Field Documentation

5.18.1.1 calc

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

5.18.1.2 Counter

```
uint32_t Counter
```

5.18.1.3 enable

```
uint16_t enable
```

5.18.1.4 fs

```
float fs
```

5.18.1.5 In

```
float In
```

5.18.1.6 Out

```
float Out
```

5.18.1.7 Pulses

```
uint32_t Pulses
```

5.18.1.8 Step

```
float Step
```

5.18.1.9 t_step

```
float t_step
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

5.19 svpwm_struct Struct Reference

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

Data Fields

- float valfa
- float vbeta
- float Ta
- float Tb
- float Tc
- void(∗ calc)()

5.19.1 Field Documentation

5.19.1.1 calc

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

5.19.1.2 Ta

```
float Ta
```

5.19.1.3 Tb

```
float Tb
```

5.19.1.4 Tc

```
float Tc
```

5.19.1.5 valfa

```
float valfa
```

5.19.1.6 vbeta

```
float vbeta
```

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

- C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_float.h

Chapter 6

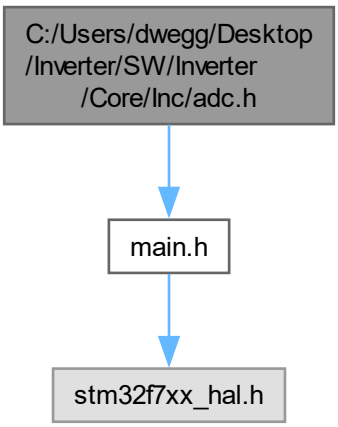
File Documentation

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

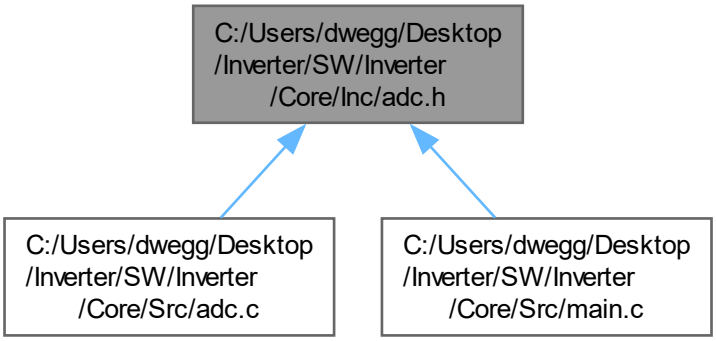
This file contains all the function prototypes for the adc.c file.

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

Include dependency graph for adc.h:



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



Functions

- void MX_ADC1_Init (void)
- void MX_ADC2_Init (void)
- void MX_ADC3_Init (void)

Variables

- ADC_HandleTypeDef hadc1
- ADC_HandleTypeDef hadc2
- ADC_HandleTypeDef hadc3

6.1.1 Detailed Description

This file contains all the function prototypes for the adc.c file.

Attention

6.1.2 Function Documentation

6.1.2.1 MX_ADC1_Init()

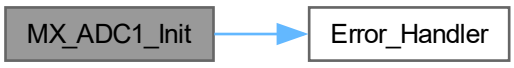
```
void MX_ADC1_Init (
             void  )
```

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

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

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

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



Here is the caller graph for this function:

6.1.2.2 MX_ADC2_Init()

```
void MX_ADC2_Init (
             void  )
```

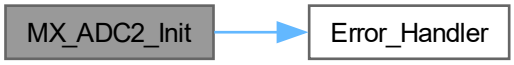
Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of conversion)

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

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

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

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



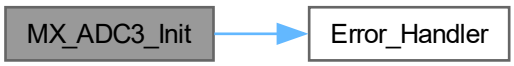
Here is the caller graph for this function:

6.1.2.3 MX_ADC3_Init()

```
void MX_ADC3_Init (
             void  )
```

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

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



Here is the caller graph for this function:

6.1.3 Variable Documentation

6.1.3.1 hadc1

```
ADC_HandleTypeDef hadc1  [extern]
```

6.1.3.2 hadc2

```
ADC_HandleTypeDef hadc2  [extern]
```

6.1.3.3 hadc3

```
ADC_HandleTypeDef hadc3  [extern]
```

6.2 adc.h

Go to the documentation of this file.

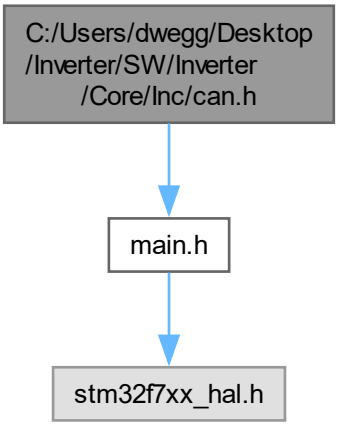
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __ADC_H__
00022 #define __ADC_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes ------------------------------------------------------------------*/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

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

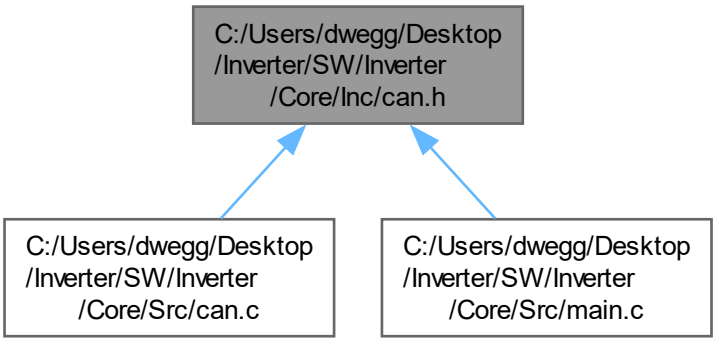
This file contains all the function prototypes for the can.c file.

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

Include dependency graph for can.h:



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



Functions

- void MX_CAN1_Init (void)

Variables

- CAN_HandleTypeDef hcan1

6.3.1 Detailed Description

This file contains all the function prototypes for the can.c file.

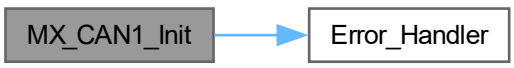
Attention

6.3.2 Function Documentation

6.3.2.1 MX_CAN1_Init()

```
void MX_CAN1_Init (
              void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:

6.3.3 Variable Documentation

6.3.3.1 hcan1

```
CAN_HandleTypeDef hcan1  [extern]
```

6.4 can.h

Go to the documentation of this file.

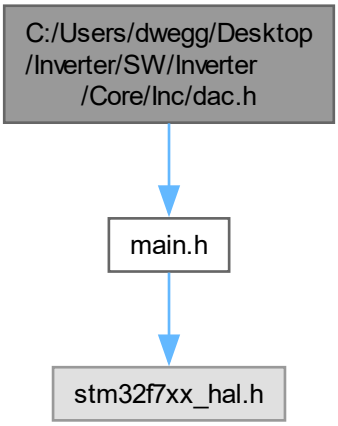
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __CAN_H__
00022 #define __CAN_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes ------------------------------------------------------------------*/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

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

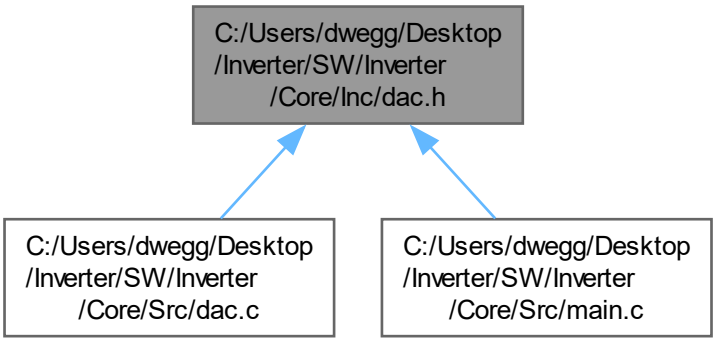
This file contains all the function prototypes for the dac.c file.

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

Include dependency graph for dac.h:



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



Functions

- void MX_DAC_Init (void)

Variables

- DAC_HandleTypeDef hdac

6.5.1 Detailed Description

This file contains all the function prototypes for the dac.c file.

Attention

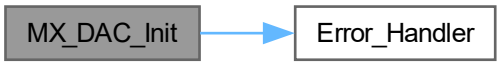
6.5.2 Function Documentation

6.5.2.1 MX_DAC_Init()

```
void MX_DAC_Init (
             void  )
```

DAC Initialization

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



Here is the caller graph for this function:

6.5.3 Variable Documentation

6.5.3.1 hdac

```
DAC_HandleTypeDef hdac  [extern]
```

6.6 dac.h

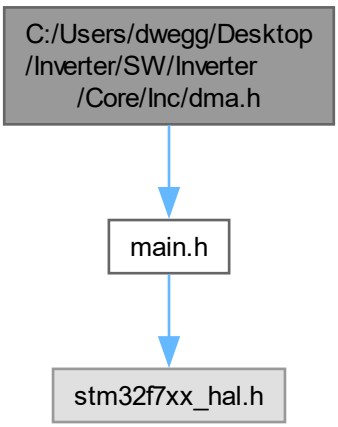
Go to the documentation of this file.

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

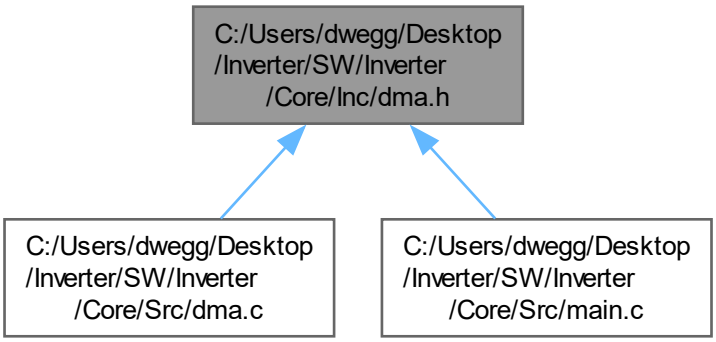
6.7 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/dma.h File Reference

This file contains all the function prototypes for the dma.c file.

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



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



Functions

- void MX_DMA_Init (void)

6.7.1 Detailed Description

This file contains all the function prototypes for the dma.c file.

Attention

6.7.2 Function Documentation

6.7.2.1 MX_DMA_Init()

```
void MX_DMA_Init (
            void  )
```

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

6.8 dma.h

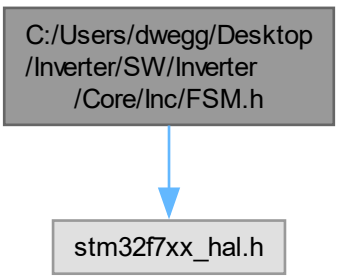
Go to the documentation of this file.

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

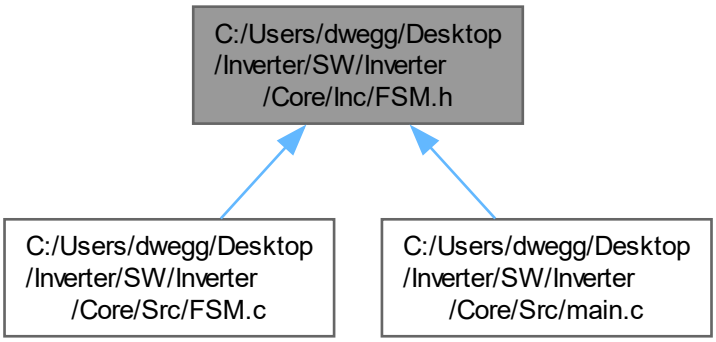
6.9 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/FSM.h File Reference

Header file for Finite State Machine (FSM) control.

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



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



Data Structures

- struct InverterOperation

  Inverter operation structure.

Enumerations

- enum InverterOperationState { INV_STATE_IDLE , INV_STATE_STARTUP , INV_STATE_RUNNING , INV_STATE_FAULT }

  Enumeration of inverter operation states.

Functions

- void inv_init (InverterOperation *inv)

  Initialize the inverter operation structure.
- void inv_FSM (InverterOperation *inv)

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

6.9.1 Detailed Description

Header file for Finite State Machine (FSM) control.

Attention

6.9.2 Enumeration Type Documentation

6.9.2.1 InverterOperationState

```
enum InverterOperationState
```

Enumeration of inverter operation states.

Enumerator

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

6.9.3 Function Documentation

6.9.3.1 inv_FSM()

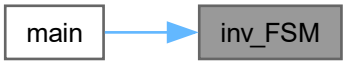
```
void inv_FSM (
            InverterOperation * inv )
```

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

Parameters

inv	Pointer to the inverter operation structure.

Here is the caller graph for this function:



6.9.3.2 inv_init()

```
void inv_init (
            InverterOperation * inv )
```

Initialize the inverter operation structure.

Parameters

inv	Pointer to the inverter operation structure.

6.10 FSM.h

Go to the documentation of this file.

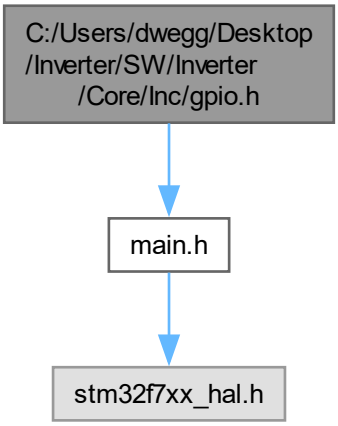
```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020 #ifndef FSM_H
00021 #define FSM_H
00022
00023 #include "stm32f7xx_hal.h"
00024
00028 typedef enum {
00029     INV_STATE_IDLE,
00030     INV_STATE_STARTUP,
00031     INV_STATE_RUNNING,
00032     INV_STATE_FAULT
00033 } InverterOperationState;
00034
00038 typedef struct {
00039     InverterOperationState state;
00041     // GPIO pins for LEDs
00042     GPIO_TypeDef *LED_port;
00043     uint16_t LED_pin;
00045     // GPIO pin for enable/disable
00046     GPIO_TypeDef *enable_port;
00047     uint16_t enable_pin;
00049     // Add any other inverter-specific variables here
00050 } InverterOperation;
00051
00057 void inv_init(InverterOperation *inv);
00058
00064 void inv_FSM(InverterOperation *inv);
00065
00066 #endif /* FSM_H */
```

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

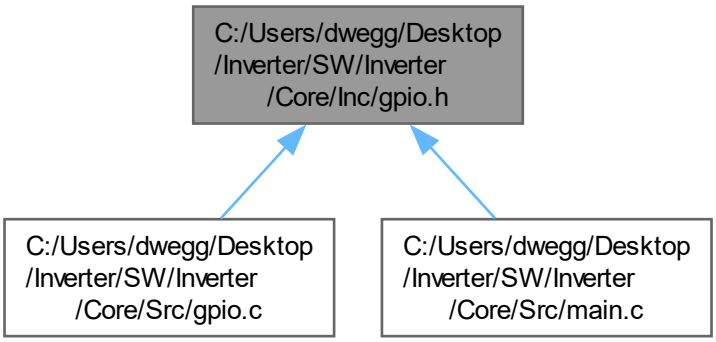
This file contains all the function prototypes for the gpio.c file.

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

Include dependency graph for gpio.h:



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



Functions

- void MX_GPIO_Init (void)

6.11.1 Detailed Description

This file contains all the function prototypes for the gpio.c file.

Attention

6.11.2 Function Documentation

6.11.2.1 MX_GPIO_Init()

```
void MX_GPIO_Init (
             void  )
```

Configure pins as Analog Input Output EVENT_OUT EXTI PA9 ------> USB_OTG_FS_VBUS PA10 ------> USB↩ _OTG_FS_ID PA11 ------> USB_OTG_FS_DM PA12 ------> USB_OTG_FS_DP Here is the caller graph for this function:

6.12 gpio.h

Go to the documentation of this file.

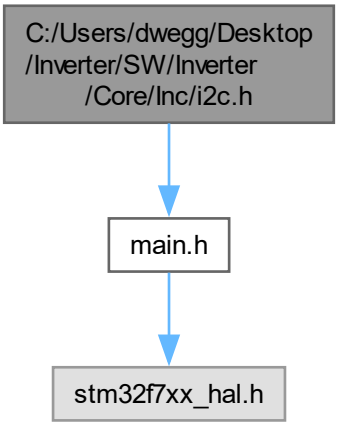
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __GPIO_H__
00022 #define __GPIO_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes ------------------------------------------------------------------*/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

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

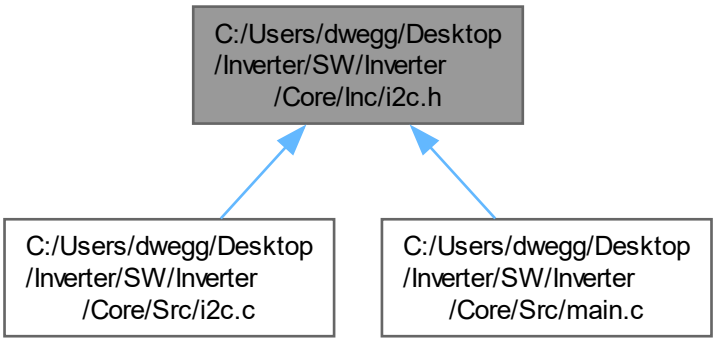
This file contains all the function prototypes for the i2c.c file.

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

Include dependency graph for i2c.h:



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



Functions

- void MX_I2C1_Init (void)

Variables

- I2C_HandleTypeDef hi2c1

6.13.1 Detailed Description

This file contains all the function prototypes for the i2c.c file.

Attention

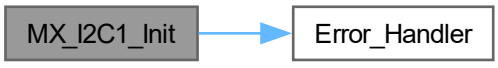
6.13.2 Function Documentation

6.13.2.1 MX_I2C1_Init()

```
void MX_I2C1_Init (
             void  )
```

Configure Analogue filter

Configure Digital filterHere is the call graph for this function:



Here is the caller graph for this function:

6.13.3 Variable Documentation

6.13.3.1 hi2c1

```
I2C_HandleTypeDef hi2c1  [extern]
```

6.14 i2c.h

Go to the documentation of this file.

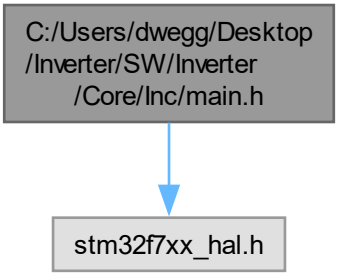
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __I2C_H__
00022 #define __I2C_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes ------------------------------------------------------------------*/
00029 #include "main.h"
00030
00031 /* USER CODE BEGIN Includes */
00032
00033 /* USER CODE END Includes */
00034
00035 extern I2C_HandleTypeDef hi2c1;
00036
00037 /* USER CODE BEGIN Private defines */
00038
00039 /* USER CODE END Private defines */
00040
00041 void MX_I2C1_Init(void);
00042
00043 /* USER CODE BEGIN Prototypes */
00044
00045 /* USER CODE END Prototypes */
00046
00047 #ifdef __cplusplus
00048 }
00049 #endif
00050
00051 #endif /* __I2C_H__ */
00052
```

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

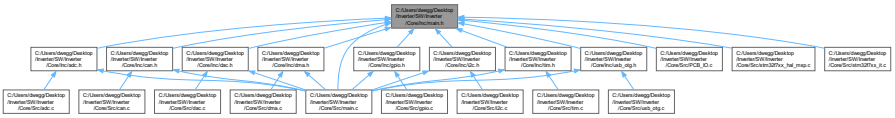
: Header for main.c file. This file contains the common defines of the application.

```
#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.

6.15.1 Detailed Description

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

Attention

6.15.2 Macro Definition Documentation

6.15.2.1 A_L_GPIO_Port

```
#define A_L_GPIO_Port GPIOD
```

6.15.2.2 A_L_Pin

```
#define A_L_Pin GPIO_PIN_12
```

6.15.2.3 A_R_GPIO_Port

```
#define A_R_GPIO_Port GPIOA
```

6.15.2.4 A_R_Pin

```
#define A_R_Pin GPIO_PIN_15
```

6.15.2.5 B_L_GPIO_Port

```
#define B_L_GPIO_Port GPIOD
```

6.15.2.6 B_L_Pin

```
#define B_L_Pin GPIO_PIN_14
```

6.15.2.7 B_R_GPIO_Port

```
#define B_R_GPIO_Port GPIOB
```

6.15.2.8 B_R_Pin

```
#define B_R_Pin GPIO_PIN_10
```

6.15.2.9 DAC_GPIO_Port

```
#define DAC_GPIO_Port GPIOA
```

6.15.2.10 DAC_Pin

```
#define DAC_Pin GPIO_PIN_4
```

6.15.2.11 DIR_GPIO_Port

```
#define DIR_GPIO_Port GPIOD
```

6.15.2.12 DIR_Pin

```
#define DIR_Pin GPIO_PIN_3
```

6.15.2.13 ENABLE_L_GPIO_Port

```
#define ENABLE_L_GPIO_Port GPIOE
```

6.15.2.14 ENABLE_L_Pin

```
#define ENABLE_L_Pin GPIO_PIN_7
```

6.15.2.15 ENABLE_R_GPIO_Port

```
#define ENABLE_R_GPIO_Port GPIOB
```

6.15.2.16 ENABLE_R_Pin

```
#define ENABLE_R_Pin GPIO_PIN_2
```

6.15.2.17 ia_L_GPIO_Port

```
#define ia_L_GPIO_Port GPIOA
```

6.15.2.18 ia_L_Pin

```
#define ia_L_Pin GPIO_PIN_0
```

6.15.2.19 ia_R_GPIO_Port

```
#define ia_R_GPIO_Port GPIOA
```

6.15.2.20 ia_R_Pin

```
#define ia_R_Pin GPIO_PIN_6
```

6.15.2.21 ib_L_GPIO_Port

```
#define ib_L_GPIO_Port GPIOA
```

6.15.2.22 ib_L_Pin

```
#define ib_L_Pin GPIO_PIN_1
```

6.15.2.23 ib_R_GPIO_Port

```
#define ib_R_GPIO_Port GPIOA
```

6.15.2.24 ib_R_Pin

```
#define ib_R_Pin GPIO_PIN_7
```

6.15.2.25 ic_L_GPIO_Port

```
#define ic_L_GPIO_Port GPIOA
```

6.15.2.26 ic_L_Pin

```
#define ic_L_Pin GPIO_PIN_2
```

6.15.2.27 ic_R_GPIO_Port

```
#define ic_R_GPIO_Port GPIOB
```

6.15.2.28 ic_R_Pin

```
#define ic_R_Pin GPIO_PIN_0
```

6.15.2.29 LED_ERR_GPIO_Port

```
#define LED_ERR_GPIO_Port GPIOD
```

6.15.2.30 LED_ERR_Pin

```
#define LED_ERR_Pin GPIO_PIN_6
```

6.15.2.31 LED_LEFT_GPIO_Port

```
#define LED_LEFT_GPIO_Port GPIOD
```

6.15.2.32 LED_LEFT_Pin

```
#define LED_LEFT_Pin GPIO_PIN_4
```

6.15.2.33 LED_RIGHT_GPIO_Port

```
#define LED_RIGHT_GPIO_Port GPIOD
```

6.15.2.34 LED_RIGHT_Pin

```
#define LED_RIGHT_Pin GPIO_PIN_5
```

6.15.2.35 PWM1_L_GPIO_Port

```
#define PWM1_L_GPIO_Port GPIOE
```

6.15.2.36 PWM1_L_Pin

```
#define PWM1_L_Pin GPIO_PIN_8
```

6.15.2.37 PWM1_R_GPIO_Port

```
#define PWM1_R_GPIO_Port GPIOA
```

6.15.2.38 PWM1_R_Pin

```
#define PWM1_R_Pin GPIO_PIN_5
```

6.15.2.39 PWM2_L_GPIO_Port

```
#define PWM2_L_GPIO_Port GPIOE
```

6.15.2.40 PWM2_L_Pin

```
#define PWM2_L_Pin GPIO_PIN_9
```

6.15.2.41 PWM2_R_GPIO_Port

```
#define PWM2_R_GPIO_Port GPIOC
```

6.15.2.42 PWM2_R_Pin

```
#define PWM2_R_Pin GPIO_PIN_6
```

6.15.2.43 PWM3_L_GPIO_Port

```
#define PWM3_L_GPIO_Port GPIOE
```

6.15.2.44 PWM3_L_Pin

```
#define PWM3_L_Pin GPIO_PIN_10
```

6.15.2.45 PWM3_R_GPIO_Port

```
#define PWM3_R_GPIO_Port GPIOB
```

6.15.2.46 PWM3_R_Pin

```
#define PWM3_R_Pin GPIO_PIN_14
```

6.15.2.47 PWM4_L_GPIO_Port

```
#define PWM4_L_GPIO_Port GPIOE
```

6.15.2.48 PWM4_L_Pin

```
#define PWM4_L_Pin GPIO_PIN_11
```

6.15.2.49 PWM4_R_GPIO_Port

```
#define PWM4_R_GPIO_Port GPIOC
```

6.15.2.50 PWM4_R_Pin

```
#define PWM4_R_Pin GPIO_PIN_7
```

6.15.2.51 PWM5_L_GPIO_Port

```
#define PWM5_L_GPIO_Port GPIOE
```

6.15.2.52 PWM5_L_Pin

```
#define PWM5_L_Pin GPIO_PIN_12
```

6.15.2.53 PWM5_R_GPIO_Port

```
#define PWM5_R_GPIO_Port GPIOB
```

6.15.2.54 PWM5_R_Pin

```
#define PWM5_R_Pin GPIO_PIN_15
```

6.15.2.55 PWM6_L_GPIO_Port

```
#define PWM6_L_GPIO_Port GPIOE
```

6.15.2.56 PWM6_L_Pin

```
#define PWM6_L_Pin GPIO_PIN_13
```

6.15.2.57 PWM6_R_GPIO_Port

```
#define PWM6_R_GPIO_Port GPIOC
```

6.15.2.58 PWM6_R_Pin

```
#define PWM6_R_Pin GPIO_PIN_8
```

6.15.2.59 SC_det_GPIO_Port

```
#define SC_det_GPIO_Port GPIOC
```

6.15.2.60 SC_det_Pin

```
#define SC_det_Pin GPIO_PIN_4
```

6.15.2.61 Tinv_L_GPIO_Port

```
#define Tinv_L_GPIO_Port GPIOC
```

6.15.2.62 Tinv_L_Pin

```
#define Tinv_L_Pin GPIO_PIN_0
```

6.15.2.63 Tinv_R_GPIO_Port

```
#define Tinv_R_GPIO_Port GPIOC
```

6.15.2.64 Tinv_R_Pin

```
#define Tinv_R_Pin GPIO_PIN_1
```

6.15.2.65 Tmot_L_GPIO_Port

```
#define Tmot_L_GPIO_Port GPIOC
```

6.15.2.66 Tmot_L_Pin

```
#define Tmot_L_Pin GPIO_PIN_2
```

6.15.2.67 Tmot_R_GPIO_Port

```
#define Tmot_R_GPIO_Port GPIOC
```

6.15.2.68 Tmot_R_Pin

```
#define Tmot_R_Pin GPIO_PIN_3
```

6.15.2.69 TRIP_L_GPIO_Port

```
#define TRIP_L_GPIO_Port GPIOA
```

6.15.2.70 TRIP_L_Pin

```
#define TRIP_L_Pin GPIO_PIN_8
```

6.15.2.71 TRIP_R_GPIO_Port

```
#define TRIP_R_GPIO_Port GPIOC
```

6.15.2.72 TRIP_R_Pin

```
#define TRIP_R_Pin GPIO_PIN_9
```

6.15.2.73 VDC_L_GPIO_Port

```
#define VDC_L_GPIO_Port GPIOA
```

6.15.2.74 VDC_L_Pin

```
#define VDC_L_Pin GPIO_PIN_3
```

6.15.2.75 VDC_R_GPIO_Port

```
#define VDC_R_GPIO_Port GPIOB
```

6.15.2.76 VDC_R_Pin

```
#define VDC_R_Pin GPIO_PIN_1
```

6.15.2.77 WRN_L_GPIO_Port

```
#define WRN_L_GPIO_Port GPIOE
```

6.15.2.78 WRN_L_Pin

```
#define WRN_L_Pin GPIO_PIN_14
```

6.15.2.79 WRN_R_GPIO_Port

```
#define WRN_R_GPIO_Port GPIOE
```

6.15.2.80 WRN_R_Pin

```
#define WRN_R_Pin GPIO_PIN_15
```

6.15.2.81 Z_L_GPIO_Port

```
#define Z_L_GPIO_Port GPIOD
```

6.15.2.82 Z_L_Pin

```
#define Z_L_Pin GPIO_PIN_15
```

6.15.2.83 Z_R_GPIO_Port

```
#define Z_R_GPIO_Port GPIOB
```

6.15.2.84 Z_R_Pin

```
#define Z_R_Pin GPIO_PIN_11
```

6.15.3 Function Documentation

6.15.3.1 Error_Handler()

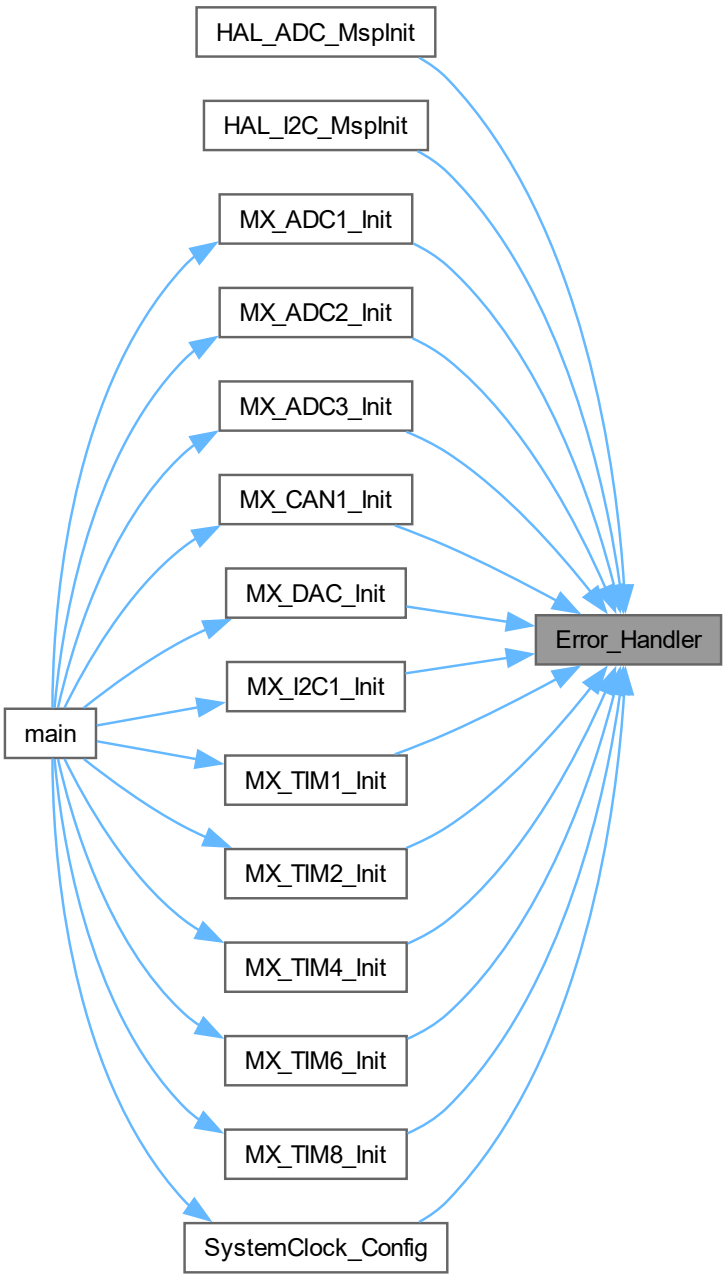
```
void Error_Handler (
            void  )
```

This function is executed in case of error occurrence.

Return values

None	

Here is the caller graph for this function:



6.16 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
```

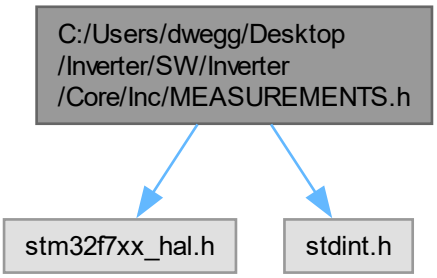
```
00105 #define PWM6_L_GPIO_Port GPIOE
00106 #define WRN_L_Pin GPIO_PIN_14
00107 #define WRN_L_GPIO_Port GPIOE
00108 #define WRN_R_Pin GPIO_PIN_15
00109 #define WRN_R_GPIO_Port GPIOE
00110 #define B_R_Pin GPIO_PIN_10
00111 #define B_R_GPIO_Port GPIOB
00112 #define Z_R_Pin GPIO_PIN_11
00113 #define 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 #ifdef __cplusplus
00150 }
00151 #endif
00152
00153 #endif /* __MAIN_H */
```

6.17 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/↩ MEASUREMENTS.h File Reference

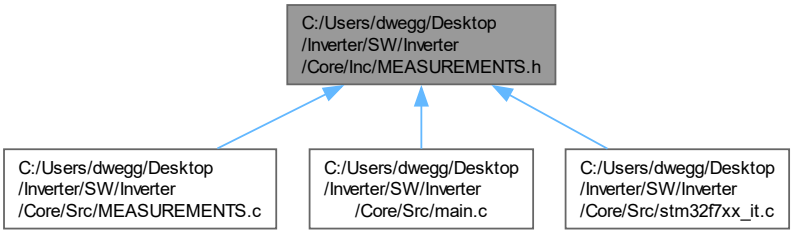
Header file for handling measurements.

```
#include "stm32f7xx_hal.h"
#include <stdint.h>
```

Include dependency graph for MEASUREMENTS.h:



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



Data Structures

- struct Encoder
- struct Measurements

Macros

- #define CURRENT_SLOPE 54.4217687f
- #define CURRENT_OFFSET 1.70068027211f
- #define VOLTAGE_SLOPE 0.003796f
- #define VOLTAGE_OFFSET 0.02083f

Functions

- uint8_t getADCelec (volatile uint32_t *ADC_raw, volatile Encoder *encoder, volatile Measurements *measurements)

  Get electrical ADC measurements.
- float getLinear (uint32_t bits, float slope, float offset)

  Convert ADC reading to physical measurement with linear response.

Variables

- volatile uint32_t ADC_LEFT_raw [4]

  Raw ADC readings for the left inverter.
- volatile uint32_t ADC_RIGHT_raw [4]

  Raw ADC readings for the right inverter.
- volatile Encoder encoder_LEFT

  Encoder data for the left inverter.
- volatile Encoder encoder_RIGHT

  Encoder data for the right inverter.
- volatile Measurements measurements_LEFT

  Measurements data for the left inverter.
- volatile Measurements measurements_RIGHT

  Measurements data for the right inverter.

6.17.1 Detailed Description

Header file for handling measurements.

Attention

6.17.2 Macro Definition Documentation

6.17.2.1 CURRENT_OFFSET

```
#define CURRENT_OFFSET 1.70068027211f
```

[V] (10/(4.7+10))∗ 2.5 V

6.17.2.2 CURRENT_SLOPE

```
#define CURRENT_SLOPE 54.4217687f
```

[A/V] (10/(4.7+10)) ∗ (1 / (12.5 mV / A))

6.17.2.3 VOLTAGE_OFFSET

```
#define VOLTAGE_OFFSET 0.02083f
```

[V] (100/(4700+100) ∗ 5 V

6.17.2.4 VOLTAGE_SLOPE

```
#define VOLTAGE_SLOPE 0.003796f
```

[V/V] 1/3 ∗ 0.011388 V

6.17.3 Function Documentation

6.17.3.1 getADCelec()

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

Get electrical ADC measurements.

Parameters

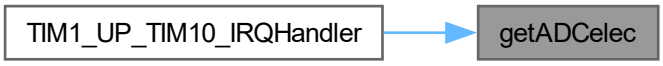
ADC_raw	Pointer to the raw ADC values array.
encoder	Pointer to the encoder struct.
measurements	Pointer to the measurements struct to store the results.

Return values

OK	0 if an error occurred, 1 if successful.

Here is the call graph for this function:

Here is the caller graph for this function:



6.17.3.2 getLinear()

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

Convert ADC reading to physical measurement with linear response.

Parameters

bits	The ADC reading.
slope	The slope (volts per unit).
offset	The offset (volts at zero).

Return values

measurement	The physical measurement.

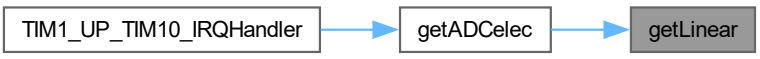
Parameters

bits	The ADC reading.
slope	The slope (units per volt).
offset	The offset (volts at zero).

Return values

measurement	The physical measurement.

Here is the caller graph for this function:



6.17.4 Variable Documentation

6.17.4.1 ADC_LEFT_raw

```
volatile uint32_t ADC_LEFT_raw[4]  [extern]
```

Raw ADC readings for the left inverter.

6.17.4.2 ADC_RIGHT_raw

```
volatile uint32_t ADC_RIGHT_raw[4]  [extern]
```

Raw ADC readings for the right inverter.

6.17.4.3 encoder_LEFT

```
volatile Encoder encoder_LEFT  [extern]
```

Encoder data for the left inverter.

6.17.4.4 encoder_RIGHT

```
volatile Encoder encoder_RIGHT  [extern]
```

Encoder data for the right inverter.

6.17.4.5 measurements_LEFT

```
volatile Measurements measurements_LEFT  [extern]
```

Measurements data for the left inverter.

6.17.4.6 measurements_RIGHT

```
volatile Measurements measurements_RIGHT  [extern]
```

Measurements data for the right inverter.

6.18 MEASUREMENTS.h

Go to the documentation of this file.

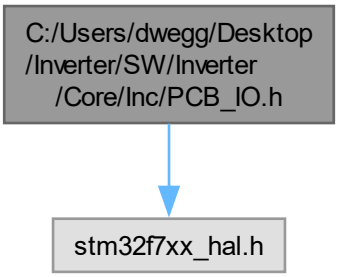
```
00001 /* USER CODE BEGIN Header */
00018 /* Define current and voltage gains/offsets */
00019 #define CURRENT_SLOPE  54.4217687f
00020 #define CURRENT_OFFSET 1.70068027211f
00021 #define VOLTAGE_SLOPE  0.003796f
00022 #define VOLTAGE_OFFSET 0.02083f
00024 /* USER CODE END Header */
00025 #ifndef MEASUREMENTS_H
00026 #define MEASUREMENTS_H
00027
00028 /* Include necessary header files */
00029 #include "stm32f7xx_hal.h"
00030 #include <stdint.h>
00031
00032 extern volatile uint32_t ADC_LEFT_raw[4];
00033 extern volatile uint32_t ADC_RIGHT_raw[4];
00034
00035
00036 /* Define encoder struct */
00037 typedef struct {
00038     uint16_t A;
00039     uint16_t B;
00040     uint16_t Z;
```

```
00041     float wm_rpm;
00042     float we;
00043     float theta_m;
00044     float theta_e;
00045     uint8_t DIR;
00046 } Encoder;
00047
00048 /* Define measurements struct */
00049 typedef struct {
00050     float ia;
00051     float ib;
00052     float ic;
00053     float VDC;
00054 } Measurements;
00055
00056 /* Declare encoder instances */
00057 extern volatile Encoder encoder_LEFT;
00058 extern volatile Encoder encoder_RIGHT;
00059
00060 /* Declare measurement instances */
00061 extern volatile Measurements measurements_LEFT;
00062 extern volatile Measurements measurements_RIGHT;
00063
00064 /* Define function prototypes */
00072 uint8_t getADCelec(volatile uint32_t* ADC_raw, volatile Encoder* encoder, volatile Measurements*
      measurements);
00080 float getLinear(uint32_t bits, float slope, float offset);
00081
00082 #endif /* MEASUREMENTS_H */
```

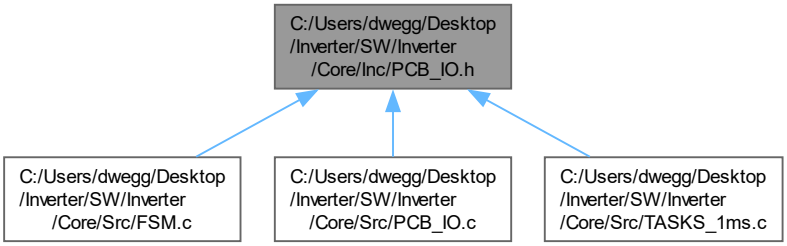
6.19 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/PCB_IO.h File Reference

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

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



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



Data Structures

- struct LED

  LED structure.

Macros

- #define SC_DET_STATE() (HAL_GPIO_ReadPin(SC_det_GPIO_Port, SC_det_Pin))
- #define DIR_STATE() (HAL_GPIO_ReadPin(DIR_GPIO_Port, DIR_Pin))
- #define WRN_STATE(port, pin) (HAL_GPIO_ReadPin(port, pin))
- #define ENABLE(port, pin) do { HAL_GPIO_WritePin(port, pin, GPIO_PIN_SET); } while(0)
- #define DISABLE(port, pin) do { HAL_GPIO_WritePin(port, pin, GPIO_PIN_RESET); } while(0)

Enumerations

- enum LEDMode { LED_MODE_BLINK_FAST , LED_MODE_BLINK_SLOW , LED_MODE_ON , LED_MODE_OFF }

Functions

- void LED_handler (LED *led, uint32_t ms_counter)

  LED handler function.

Variables

- LED led_left
- LED led_right
- LED led_error

6.19.1 Detailed Description

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

Attention

6.19.2 Macro Definition Documentation

6.19.2.1 DIR_STATE

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

6.19.2.2 DISABLE

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

6.19.2.3 ENABLE

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

6.19.2.4 SC_DET_STATE

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

6.19.2.5 WRN_STATE

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

6.19.3 Enumeration Type Documentation

6.19.3.1 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

6.19.4 Function Documentation

6.19.4.1 LED_handler()

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

LED handler function.

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

Parameters

led	Pointer to the LED structure.
ms_counter	Millisecond counter for timing.

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

Parameters

led	Pointer to the LED structure.
ms_counter	Current millisecond counter.

Here is the caller graph for this function:

6.19.5 Variable Documentation

6.19.5.1 led_error

```
LED led_error  [extern]
```

6.19.5.2 led_left

```
LED led_left  [extern]
```

6.19.5.3 led_right

```
LED led_right  [extern]
```

6.20 PCB_IO.h

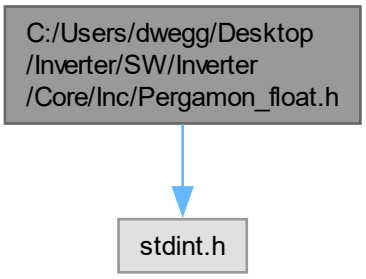
Go to the documentation of this file.

```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020
00021 #ifndef PCB_IO_H
00022 #define PCB_IO_H
00023
00024 #include "stm32f7xx_hal.h"
00025
00026 // Read SC_det and DIR GPIOs
00027 #define SC_DET_STATE()              (HAL_GPIO_ReadPin(SC_det_GPIO_Port, SC_det_Pin))
00028 #define DIR_STATE()                 (HAL_GPIO_ReadPin(DIR_GPIO_Port, DIR_Pin))
00029
00030 // Read WRN GPIOs
00031 #define WRN_STATE(port, pin)        (HAL_GPIO_ReadPin(port, pin))
00032
00033 // Control ENABLE GPIOs
00034 #define ENABLE(port, pin)           do { HAL_GPIO_WritePin(port, pin, GPIO_PIN_SET); } while(0)
00035 #define DISABLE(port, pin)          do { HAL_GPIO_WritePin(port, pin, GPIO_PIN_RESET); } while(0)
00036
00037 // Define LED modes
00038 typedef enum {
00039     LED_MODE_BLINK_FAST,
00040     LED_MODE_BLINK_SLOW,
00041     LED_MODE_ON,
00042     LED_MODE_OFF
00043 } LEDMode;
00044
00048 typedef struct {
00049     GPIO_TypeDef *port;
00050     uint16_t pin;
00051     LEDMode mode;
00052 } LED;
00053
00054 // Declare LED variables as extern
00055 extern LED led_left;
00056 extern LED led_right;
00057 extern LED led_error;
00058
00059 // Function prototypes
00068 void LED_handler(LED *led, uint32_t ms_counter);
00069
00070 #endif /* PCB_IO_H */
```

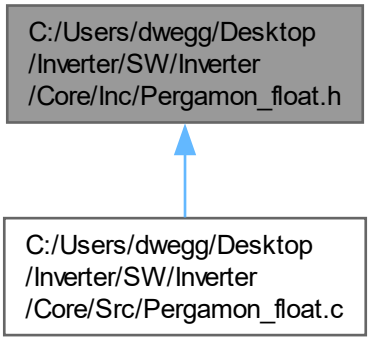
6.21 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/Pergamon_↩ float.h File Reference

```
#include "stdint.h"
```

Include dependency graph for Pergamon_float.h:



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



Data Structures

- struct pi_aw_struct
- struct pi_struct
- struct clarke3F_struct
- struct iclarke3F_struct
- struct rot_struct
- struct irot_struct
- struct angle_struct
- struct svpwm_struct

- struct rampa_struct
- struct rampa_dual_struct
- struct datalog_struct
- struct avg_struct_10
- struct RMS_struct
- struct filtreLP_struct
- struct step_struct

Macros

- #define SQ2 1.4142135624F
- #define ISQ2 0.7071067812F
- #define SQ3 1.7320508076F
- #define ISQ3 0.5773502692F
- #define PI 3.1415926536F
- #define IPI 0.3183098862F
- #define PI2 6.2831853072F
- #define IPI2 0.1591549431F
- #define INV_DEG 0.0027777778F
- #define INV3 0.3333333333F
- #define DIV2 0.5F
- #define PI_DEFAULTS_AW
- #define PI_DEFAULTS
- #define PI_EXTSAT_DEFAULTS
- #define CLARKE3F_DEFAULTS
- #define ICLARKE3F_DEFAULTS
- #define ROT_DEFAULTS
- #define IROT_DEFAULTS
- #define ANGLE_DEFAULTS
- #define SVPWM_DEFAULTS
- #define RAMPA_DEFAULTS
- #define RAMPA_DUAL_DEFAULTS
- #define N_DATALOG 512
- #define DATALOG_DEFAULTS
- #define AVG_DEFAULTS
- #define RMS_DEFAULTS
- #define FILTRELP_DEFAULTS
- #define STEP_DEFAULTS

Functions

- void pi_aw_calc (volatile pi_aw_struct *v) __attribute__((section(".ccmram")))
- void pi_init (volatile pi_struct *v) __attribute__((section(".ccmram")))
- void pi_calc (volatile pi_struct *v) __attribute__((section(".ccmram")))
- void pi_extsat_calc (volatile pi_struct *v)
- void clarke3F_calc (volatile clarke3F_struct *v) __attribute__((section(".ccmram")))
- void iclarke3F_calc (volatile iclarke3F_struct *v)
- void rot_calc (volatile rot_struct *v) __attribute__((section(".ccmram")))
- void irot_calc (volatile irot_struct *v) __attribute__((section(".ccmram")))
- void angle_calc (volatile angle_struct *p) __attribute__((section(".ccmram")))
- void svpwm_calc (volatile svpwm_struct *v) __attribute__((section(".ccmram")))
- void rampa_calc (volatile rampa_struct *v) __attribute__((section(".ccmram")))
- void rampa_dual_calc (volatile rampa_dual_struct *v) __attribute__((section(".ccmram")))

- void datalog_calc (volatile datalog_struct *dl)
- void avg_calc_10_samples (volatile avg_struct_10 *v)
- void RMS_calc (volatile RMS_struct *v) __attribute__((section(".ccmram")))
- void filtreLP_init (volatile filtreLP_struct *v)
- void filtreLP_calc (volatile filtreLP_struct *v) __attribute__((section(".ccmram")))
- void step_calc (volatile step_struct *v) __attribute__((section(".ccmram")))

6.21.1 Macro Definition Documentation

6.21.1.1 ANGLE_DEFAULTS

```
#define ANGLE_DEFAULTS
```

Value:

```
{   0,  \
0,  \
0,  \
(void (*)(int32_t))angle_calc }
```

6.21.1.2 AVG_DEFAULTS

```
#define AVG_DEFAULTS
```

Value:

```
{   0,\
{0} }
```

6.21.1.3 CLARKE3F_DEFAULTS

```
#define CLARKE3F_DEFAULTS
```

Value:

```
{ 0, \
0, \
0, \
0, \
(void (*)(int32_t))clarke3F_calc }
```

6.21.1.4 DATALOG_DEFAULTS

```
#define DATALOG_DEFAULTS
```

Value:

```
{   0, \
0, \
0, \
1, \
0x00000000, /*puntero a quien sabe donde?*/ \
(void (*)(int32_t))datalog_calc }
```

6.21.1.5 DIV2

```
#define DIV2 0.5F
```

6.21.1.6 FILTRELP_DEFAULTS

```
#define FILTRELP_DEFAULTS
```

Value:

```
{   0, \
0, \
0, \
0, \
0, \
0, \
(void (*)(int32_t)) filtreLP_init, \
(void (*)(int32_t)) filtreLP_calc }
```

6.21.1.7 ICLARKE3F_DEFAULTS

```
#define ICLARKE3F_DEFAULTS
```

Value:

```
{   0, \
0, \
0, \
0, \
(void (*)(int32_t))iclarke3F_calc }
```

6.21.1.8 INV3

```
#define INV3 0.3333333333F
```

6.21.1.9 INV_DEG

```
#define INV_DEG 0.0027777778F
```

6.21.1.10 IPI

```
#define IPI 0.3183098862F
```

6.21.1.11 IPI2

```
#define IPI2 0.1591549431F
```

6.21.1.12 IROT_DEFAULTS

```
#define IROT_DEFAULTS
```

Value:

```
{   0, \
0, \
0, \
0, \
0, \
(void (*)(int32_t))irot_calc }
```

6.21.1.13 ISQ2

```
#define ISQ2 0.7071067812F
```

6.21.1.14 ISQ3

```
#define ISQ3 0.5773502692F
```

6.21.1.15 N_DATALOG

```
#define N_DATALOG 512
```

6.21.1.16 PI

```
#define PI 3.1415926536F
```

6.21.1.17 PI2

```
#define PI2 6.2831853072F
```

6.21.1.18 PI_DEFAULTS

```
#define PI_DEFAULTS
```

Value:

```
{   0, \
0, \
0, \
0, \
0, \
0, \
{0,0}, \
0, \
0, \
0, \
0, \
0, \
{0,0}, \
(void (*)(int32_t))pi_init, \
(void (*)(int32_t))pi_calc   }
```

6.21.1.19 PI_DEFAULTS_AW

```
#define PI_DEFAULTS_AW
```

Value:

```
{   0, \
0, \
0, \
0, \
0, \
{0,0}, \
0, \
0, \
0, \
0, \
0, \
{0,0}, \
{0,0}, \
(void (*)(int32_t))pi_aw_calc }
```

6.21.1.20 PI_EXTSAT_DEFAULTS

```
#define PI_EXTSAT_DEFAULTS
```

Value:

```
{   0, \
0, \
0, \
0, \
0, \
0, \
{0,0}, \
0, \
0, \
0, \
0, \
0, \
{0,0}, \
(void (*)(int32_t))pi_init, \
(void (*)(int32_t))pi_extsat_calc }
```

6.21.1.21 RAMPA_DEFAULTS

```
#define RAMPA_DEFAULTS
```

Value:

```
{   0, \
0, \
0, \
0, \
(void (*)(int32_t)) rampa_calc }
```

6.21.1.22 RAMPA_DUAL_DEFAULTS

```
#define RAMPA_DUAL_DEFAULTS
```

Value:

```
{   0, \
0, \
0, \
0, \
0, \
(void (*)(int32_t)) rampa_dual_calc }
```

6.21.1.23 RMS_DEFAULTS

```
#define RMS_DEFAULTS
```

Value:

```
{   0, \
0, \
0, \
0, \
0, \
0, \
0   }
```

6.21.1.24 ROT_DEFAULTS

```
#define ROT_DEFAULTS
```

Value:

```
{   0, \
0, \
0, \
0, \
0, \
0, \
(void (*)(int32_t))rot_calc }
```

6.21.1.25 SQ2

```
#define SQ2 1.4142135624F
```

6.21.1.26 SQ3

```
#define SQ3 1.7320508076F
```

6.21.1.27 STEP_DEFAULTS

```
#define STEP_DEFAULTS
```

Value:

```
{     0.0F, \
0.0F, \
0.0F, \
0.0F, \
0.0F, \
0, \
0, \
0, \
(void (*)(int32_t)) step_calc }
```

6.21.1.28 SVPWM_DEFAULTS

```
#define SVPWM_DEFAULTS
```

Value:

```
{   0, \
0, \
0, \
0, \
0, \
(void (*)(int32_t))svpwm_calc }
```

6.21.2 Function Documentation

6.21.2.1 angle_calc()

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

6.21.2.2 avg_calc_10_samples()

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

6.21.2.3 clarke3F_calc()

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

6.21.2.4 datalog_calc()

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

6.21.2.5 filtreLP_calc()

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

6.21.2.6 filtreLP_init()

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

6.21.2.7 iclarke3F_calc()

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

6.21.2.8 irot_calc()

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

6.21.2.9 pi_aw_calc()

```
void pi_aw_calc (
            volatile pi_aw_struct * v )
```

6.21.2.10 pi_calc()

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

6.21.2.11 pi_extsat_calc()

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

6.21.2.12 pi_init()

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

6.21.2.13 rampa_calc()

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

6.21.2.14 rampa_dual_calc()

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

6.21.2.15 RMS_calc()

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

6.21.2.16 rot_calc()

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

6.21.2.17 step_calc()

```
void step_calc (
            volatile step_struct * v )
```

6.21.2.18 svpwm_calc()

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

6.22 Pergamon_float.h

Go to the documentation of this file.

```
00001 /*
00002                 _____                                          _
00003               |  __ \                                        | |
00004               | |__) |__ _ __ __ _  __ _ _ __ ___   ___  _ __   | |__
00005               |  ___/ _ \ '__/ _` |/ _` | '_ ` _ \ / _ \| '_ \  | '_ \
00006               | |   |  __/ | | (_| | (_| | | | | | | (_) | | | |_| | | |
00007               |_|    \___|_|  \__, |\__,_|_| |_| |_|\___/|_| |_(_)_| |_|
00008                                  __/ |
00009                                 |___/
00010
00011 Arxiu que engloba totes les llibreries independents del hardware del CITCEA en una.
00012 Aquesta llibreria est• basada en l'Alexandria feta per en Gabriel Gross i en  Quim L•pez Mestre el
      Setembre del 2006
00013 Feta pel Gabriel Gross, Daniel Heredero i Tom•s Lled• el desembre de 2015.
00014 Tradu•da a floats per Lucas Bouz•n el Mar• de 2020.
00015 */
00016 #include "stdint.h"
00017
00018 #define SQ2     1.4142135624F   // sqrt(2)
00019 #define ISQ2    0.7071067812F   // 1/sqrt(2)
```

```
00020 #define SQ3     1.7320508076F   // sqrt(3)
00021 #define ISQ3    0.5773502692F   // 1/sqrt(3)
00022 #define PI      3.1415926536F   // Pi
00023 #define IPI     0.3183098862F   // 1/Pi
00024 #define PI2     6.2831853072F   // 2*Pi
00025 #define IPI2    0.1591549431F   // 1/(2*Pi)
00026 #define INV_DEG 0.0027777778F   // 1/360
00027 #define INV3    0.3333333333F   // 1/3
00028 #define DIV2    0.5F            // 1/2
00029
00030 //PI
00031 //=====================================================================
00032 // Inclou saturaci\x00f3 interna amb antiwindup i ffw
00033
00034 #define PI_DEFAULTS_AW {    0, \
00035                         0, \
00036                         0, \
00037                         0, \
00038                         0, \
00039                         {0,0}, \
00040                         0, \
00041                         0, \
00042                         0, \
00043                         0, \
00044                         0, \
00045                         {0,0}, \
00046                         {0,0}, \
00047                         (void (*)(int32_t))pi_aw_calc   }
00048
00049
00050 typedef struct
00051 {
00052     uint16_t        enable;       // si enable=1 fa el pid, si no pid_out = 0
00053     float           Ts;           // Per\x00f2ode d'execuci\x00f2
00054     float           Kp;           // Kp
00055     float           Ki;           // Ki
00056     float           Kaw;     // Kaw
00057     float           e[2];       // Error k i k-1
00058     float           pi_consig; // consigna
00059     float           pi_fdb;       // realimentaci\x00f2
00060     float           pi_out_max;   // Maximum output
00061     float           pi_out_min;   // Minimum output
00062     float           pi_out_presat;
00063     float           pi_out_postsat;
00064     float           pi_out;       // PID output
00065     float           pi_int[2];    // Integrator Part
00066     float           pi_ffw[2];  // feedforward k i k-1
00067     void(*calc)();          // Punter a la funci\x00f2 calc
00068 } pi_aw_struct;
00069
00070 void pi_aw_calc(volatile pi_aw_struct *v) __attribute__( ( section ( ".ccmram" ) ) ); //__attribute__(
      ( section ( ".code_in_ram" ) ) );
00071
00072 //PI
00073 //=====================================================================
00074 // Inclou saturaci\x00f2 interna amb antiwindup i ffw
00075
00076 #define PI_DEFAULTS {   0, \
00077                         0, \
00078                         0, \
00079                         0, \
00080                         0, \
00081                         0, \
00082                         {0,0}, \
00083                         0, \
00084                         0, \
00085                         0, \
00086                         0, \
00087                         0, \
00088                         {0,0}, \
00089                         (void (*)(int32_t))pi_init, \
00090                         (void (*)(int32_t))pi_calc  }
00091
00092 // Saturaci\x00f2 externa, inclou ffw
00093 #define PI_EXTSAT_DEFAULTS {    0, \
00094                                 0, \
00095                                 0, \
00096                                 0, \
00097                                 0, \
00098                                 0, \
00099                                 {0,0}, \
00100                                 0, \
00101                                 0, \
00102                                 0, \
00103                                 0, \
00104                                 0, \
00105                                 {0,0}, \
```

```
00106                                (void (*)(int32_t))pi_init, \
00107                                (void (*)(int32_t))pi_extsat_calc   }
00108
00109 typedef struct
00110 {
00111     uint16_t        enable;       // si enable=1 fa el pid, si no pid_out = 0
00112     float           Ts;           // Per?ode d'execuci?
00113     float           Kp;           // Kp
00114     float           Ki;           // Ki
00115     float           K0;           // K0 = Kp + (Ts*Ki)/2
00116     float           K1;           // K0 = -Kp + (Ts*Ki)/2
00117     float           e[2];       // Error k i k-1
00118     float           pi_consig; // consigna
00119     float           pi_fdb;       // realimentaci?
00120     float           pi_out_max;   // Maximum output
00121     float           pi_out_min;   // Minimum output
00122     float           pi_out;       // PID output
00123     float           pi_ffw[2];  // feedforward k i k-1
00124     void(*init)();          // Punter a la funci? init
00125     void(*calc)();          // Punter a la funci? calc
00126 } pi_struct;
00127
00128 void pi_init(volatile pi_struct *v) __attribute__( ( section ( ".ccmram" ) ) ); //__attribute__( (
      section ( ".code_in_ram" ) ) );
00129 void pi_calc(volatile pi_struct *v) __attribute__( ( section ( ".ccmram" ) ) ); //__attribute__( (
      section ( ".code_in_ram" ) ) );
00130 void pi_extsat_calc(volatile pi_struct *v);
00131
00132
00133 //CLARKE 3F
00134 //=====================================================================
00135 //La funci? Clarke3F projecta per un sistema trifasic sense neutre A i, B en el pla de Park
      Alfa(D)-Beta(Q)
00136 //Entrades amb components simples
00137
00138 #define CLARKE3F_DEFAULTS { 0, \
00139                             0, \
00140                             0, \
00141                             0, \
00142                             (void (*)(int32_t))clarke3F_calc }
00143
00144 typedef struct
00145 {
00146     float           a;          //entrada eix A
00147     float           b;          //entrada eix B
00148     float           D;          //sortida eix D
00149     float           Q;          //sortida eix Q
00150     void(*calc)();
00151 } clarke3F_struct;
00152 void clarke3F_calc(volatile clarke3F_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
      //__attribute__( ( section ( ".code_in_ram" ) ) );
00153
00154 //ICLARKE 3F
00155 //=====================================================================
00156 //La funci? iClarke3F projecta per un sistema trifasic sense neutre Alfa i Beta en el pla de natural
      ABC
00157 //Sortides amb components simples
00158
00159 #define ICLARKE3F_DEFAULTS {    0, \
00160                                 0, \
00161                                 0, \
00162                                 0, \
00163                                 (void (*)(int32_t))iclarke3F_calc }
00164
00165 typedef struct
00166 {
00167     float           D;          //entrada eix D
00168     float           Q;          //entrada eix Q
00169     float           a;          //sortida eix A
00170     float           b;          //sortida eix B
00171     void(*calc)();
00172 } iclarke3F_struct;
00173 void iclarke3F_calc(volatile iclarke3F_struct *v);
00174
00175 //Rotacio (la negativa, ergo sentit horari) - PARKE??
00176 //=====================================================================
00177 //La funcion Rotacio fa rotar els eixos de Clarke Alfa (D) i Beta (Q) en funci? de sinFi i cosFi
00178 //ens donen les coordenades est?tiques (refer?ncia sincrona) d i q
00179
00180 #define ROT_DEFAULTS    {   0, \
00181                             0, \
00182                             0, \
00183                             0, \
00184                             0, \
00185                             0, \
00186                             (void (*)(int32_t))rot_calc }
00187
```

```
00188 typedef struct
00189 {
00190     float           D;          //eix Alfa (D) d'entrada
00191     float           Q;          //eix Beta (Q) d'entrada
00192     float           sinFi;      //sinus de l'angle que es vol girar
00193     float           cosFi;      //cosinus de l'angle que es vol girar
00194     float           d;          //eix d girat
00195     float           q;          //eix q girat
00196     void(*calc)();
00197 } rot_struct;
00198 void rot_calc(volatile rot_struct *v) __attribute__( ( section ( ".ccmram" ) ) ); //__attribute__( (
      section ( ".code_in_ram" ) ) );
00199
00200 //Rotacio inversa (la positiva, ergo sentit antihorari) - IPARKE??
00201 //=============================================================================
00202 //La funci iPark def la rotaci dels eixos de Park a en funci de sinFi i cosFi
00203 //ens donen les coordenades rotatories Alfa (D) - Beta (Q)
00204
00205 #define IROT_DEFAULTS   {   0, \
00206                             0, \
00207                             0, \
00208                             0, \
00209                             0, \
00210                             0, \
00211                             (void (*)(int32_t))irot_calc }
00212
00213 typedef struct
00214 {
00215     float           d;          //eix d d'entrada
00216     float           q;          //eix q d'entrada
00217     float           sinFi;      //sinus de l'angle que es vol girar
00218     float           cosFi;      //cosinus de l'angle que es vol girar
00219     float           D;          //eix Alfa (D) de sortida
00220     float           Q;          //eix Beta (Q) de sortida girat
00221     void(*calc)();
00222 } irot_struct;
00223 void irot_calc(volatile irot_struct *v) __attribute__( ( section ( ".ccmram" ) ) ); //__attribute__( (
      section ( ".code_in_ram" ) ) );
00224
00225 //Generaci angle
00226 //=============================================================================
00227 // Genera un angle a partir d'una freq��ncia fixa
00228
00229 #define ANGLE_DEFAULTS {    0,  \
00230                             0,  \
00231                             0,  \
00232                             (void (*)(int32_t))angle_calc }
00233
00234 typedef struct {
00235     float       freq;           // freq��ncia de la xarxa. Sortida del 1er filtre de freq��ncia.
00236     float       Ts;               // freq��ncia d'execuci de la integracio de la freq��ncia
00237     float       angle;          // angle de la xarxa.
00238 //  float       angle_iq32;     // angle de la xarxa.
00239     void(*calc)();              //Punter a la funci init
00240 } angle_struct;
00241 void angle_calc(volatile angle_struct *p) __attribute__( ( section ( ".ccmram" ) ) ); //__attribute__(
      ( section ( ".code_in_ram" ) ) );
00242
00243 //SVPWM
00244 //=============================================================================
00245 //Permet generar les 3 sortides SVPWM (Ta, Tb i Tc)
00246 //Partint de dues entrades en tant per 1 (Ualfa, Ubeta)
00247
00248 #define SVPWM_DEFAULTS  {   0, \
00249                             0, \
00250                             0, \
00251                             0, \
00252                             0, \
00253                             (void (*)(int32_t))svpwm_calc   }
00254
00255 typedef struct
00256 {
00257     float           valfa;      //Entrada en tant per 1
00258     float           vbeta; //Entrada en tant per 1
00259     float           Ta;         //Sortida 0-1
00260     float           Tb;         //Sortida 0-1
00261     float           Tc;         //Sortida 0-1
00262     void(*calc)();
00263 } svpwm_struct;
00264 void svpwm_calc(volatile svpwm_struct *v) __attribute__( ( section ( ".ccmram" ) ) ); //__attribute__(
      ( section ( ".code_in_ram" ) ) );
00265
00266 // RAMPA
00267 //=============================================================================
00268
00269 #define RAMPA_DEFAULTS  {   0, \
00270                             0, \
```

```
00271                               0, \
00272                               0, \
00273                               (void (*)(int32_t)) rampa_calc }
00274 typedef struct
00275 {
00276     float       in;
00277     float       out;
00278     float       Incr;
00279     uint8_t     enable;
00280     void(*calc)();
00281 } rampa_struct;
00282
00283 #define RAMPA_DUAL_DEFAULTS {   0, \
00284                               0, \
00285                               0, \
00286                               0, \
00287                               0, \
00288                               (void (*)(int32_t)) rampa_dual_calc }
00289
00290 typedef struct
00291 {
00292     float       in;
00293     float       out;
00294     float       Incr;
00295     float       Decr;
00296     uint8_t     enable;
00297     void(*calc)();
00298 } rampa_dual_struct;
00299 void rampa_calc(volatile rampa_struct *v) __attribute__( ( section ( ".ccmram" ) ) ); //__attribute__(
      ( section ( ".code_in_ram" ) ) );
00300 void rampa_dual_calc(volatile rampa_dual_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00301
00302 //DATALOG
00303 //================================================================================
00304
00305 #define N_DATALOG 512
00306 #define DATALOG_DEFAULTS    {   0, \
00307                                 0, \
00308                                 0, \
00309                                 1, \
00310                                 0x00000000, /*�puntero a quien sabe donde?*/ \
00311                                 (void (*)(int32_t))datalog_calc }
00312                                 // (!) no se inicializa el log
00313 typedef struct
00314 {
00315
00316     uint16_t        i;      //varible d'estat
00317     uint16_t        j;      //varible d'estat
00318     uint16_t        estat; //0=parat, 1=inici, 2=running
00319     uint16_t        prescaler;
00320     float           *var;
00321     void(*calc)();          // (!) la funci�n calc no se pone lo �ltimo ya que en DATALOG_DEFAULTS lo
      �ltimo tiene que ser  el log (no se inicializa)
00322     float           log[N_DATALOG];
00323 } datalog_struct;
00324 void datalog_calc(volatile datalog_struct *dl); // __attribute__( ( section ( ".ccmram" ) ) );
      //__attribute__( ( section ( ".code_in_ram" ) ) );
00325
00326 //AVG
00327 //================================================================================
00328
00329 #define AVG_DEFAULTS    {   0,\
00330                             {0} }
00331
00332 typedef struct
00333 {
00334     float       out;        // variable de sortida
00335     float       in[10];      // variable d'entrada i anteriors
00336 }avg_struct_10;
00337
00338 void avg_calc_10_samples(volatile avg_struct_10 *v);
00339
00340 //RMS
00341 //================================================================================
00342
00343 #define RMS_DEFAULTS    {   0, \
00344                             0, \
00345                             0, \
00346                             0, \
00347                             0, \
00348                             0, \
00349                             0   }
00350
00351 typedef struct {
00352     float       T_exec;     //Cada quan executes la funci�
00353     float       Measure;     //Senyal a RMSear
00354     float       Sq_Sum;     //Suma quadrats
```

```
00355     float       Out_RMS;    //Senyal RMSeat
00356     float       Freq;       //Freq??ncia de sortida de la pll
00357     float       Angle;        //angle
00358     float       Angle_ant;  //angle anterior
00359 } RMS_struct;
00360 void RMS_calc(volatile RMS_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
00361
00362 // Filtre 1er ordre
00363 //=====================================================================
00364
00365 #define FILTRELP_DEFAULTS   {   0, \
00366                                 0, \
00367                                 0, \
00368                                 0, \
00369                                 0, \
00370                                 0, \
00371                                 (void (*)(int32_t)) filtreLP_init, \
00372                                 (void (*)(int32_t)) filtreLP_calc }
00373 typedef struct
00374 {
00375     float       in;
00376     float       out;
00377     float       alfa;
00378     float       Ts;
00379     float       fc;
00380     uint16_t    enable;
00381     void(*init)();
00382     void(*calc)();
00383 } filtreLP_struct;
00384 void filtreLP_init(volatile filtreLP_struct *v);
00385 void filtreLP_calc(volatile filtreLP_struct *v) __attribute__( ( section ( ".ccmram" ) ) );
    //__attribute__( ( section ( ".code_in_ram" ) ) );;
00386
00387
00388 //Step
00389 //=====================================================================
00390 // Afegeix un step a una variable d'entrada amb l'ample i la duraci desitjada. Permet mes o menys 20 s
    a 100 kHz
00391
00392 #define STEP_DEFAULTS {     0.0F, \
00393                             0.0F, \
00394                             0.0F, \
00395                             0.0F, \
00396                             0.0F, \
00397                             0, \
00398                             0, \
00399                             0, \
00400                             (void (*)(int32_t)) step_calc }
00401 typedef struct
00402 {
00403     float           fs;         // Freqncia d'execuci de la funci.
00404     float           In;         // Variable d'entrada
00405     float           Out;        // Variable de sortida (amb el step quan calgui)
00406     float           Step;       // Amplitud del step
00407     float           t_step;     // Duraci del step en segons.
00408     uint32_t        Pulses;     // Polsos peper comptar els segons
00409     uint32_t        Counter;    // Comptador pels polsos
00410     uint16_t        enable;     // Habilita el step.
00411     void(*calc)();
00412 } step_struct;
00413 void step_calc(volatile step_struct *v) __attribute__( ( section ( ".ccmram" ) ) ); //__attribute__( (
    section ( ".code_in_ram" ) ) );;
```

6.23 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/stm32f7xx_↩ hal_conf.h File Reference

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

```
#include "stm32f7xx_hal_pwr.h"
#include "stm32f7xx_hal_tim.h"
```

Include dependency graph for stm32f7xx_hal_conf.h:

Macros

- #define HAL_MODULE_ENABLED

  This is the list of modules to be used in the HAL driver.
- #define HAL_ADC_MODULE_ENABLED
- #define HAL_CAN_MODULE_ENABLED
- #define HAL_DAC_MODULE_ENABLED
- #define HAL_TIM_MODULE_ENABLED
- #define HAL_GPIO_MODULE_ENABLED
- #define HAL_EXTI_MODULE_ENABLED
- #define HAL_DMA_MODULE_ENABLED
- #define HAL_RCC_MODULE_ENABLED
- #define HAL_FLASH_MODULE_ENABLED
- #define HAL_PWR_MODULE_ENABLED
- #define HAL_I2C_MODULE_ENABLED
- #define HAL_CORTEX_MODULE_ENABLED
- #define HSE_VALUE ((uint32_t)20000000U)

  Adjust the value of External High Speed oscillator (HSE) used in your application. This value is used by the RCC HAL module to compute the system frequency (when HSE is used as system clock source, directly or through the PLL).
- #define HSE_STARTUP_TIMEOUT ((uint32_t)100U)
- #define HSI_VALUE ((uint32_t)16000000U)

  Internal High Speed oscillator (HSI) value. This value is used by the RCC HAL module to compute the system frequency (when HSI is used as system clock source, directly or through the PLL).
- #define LSI_VALUE ((uint32_t)32000U)

  Internal Low Speed oscillator (LSI) value.
- #define LSE_VALUE ((uint32_t)32768U)

  External Low Speed oscillator (LSE) value.
- #define LSE_STARTUP_TIMEOUT ((uint32_t)5000U)
- #define EXTERNAL_CLOCK_VALUE ((uint32_t)12288000U)

  External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S_CKIN pad.
- #define VDD_VALUE 3300U

  This is the HAL system configuration section.
- #define TICK_INT_PRIORITY ((uint32_t)15U)
- #define USE_RTOS 0U
- #define PREFETCH_ENABLE 0U
- #define ART_ACCELERATOR_ENABLE 0U /* To enable instruction cache and prefetch */
- #define USE_HAL_ADC_REGISTER_CALLBACKS 0U /* ADC register callback disabled */
- #define USE_HAL_CAN_REGISTER_CALLBACKS 0U /* CAN register callback disabled */
- #define USE_HAL_CEC_REGISTER_CALLBACKS 0U /* CEC register callback disabled */
- #define USE_HAL_CRYP_REGISTER_CALLBACKS 0U /* CRYP register callback disabled */
- #define USE_HAL_DAC_REGISTER_CALLBACKS 0U /* DAC register callback disabled */
- #define USE_HAL_DCMI_REGISTER_CALLBACKS 0U /* DCMI register callback disabled */
- #define USE_HAL_DFSDM_REGISTER_CALLBACKS 0U /* DFSDM register callback disabled */

- #define USE_HAL_DMA2D_REGISTER_CALLBACKS 0U /* DMA2D register callback disabled */
- #define USE_HAL_DSI_REGISTER_CALLBACKS 0U /* DSI register callback disabled */
- #define USE_HAL_ETH_REGISTER_CALLBACKS 0U /* ETH register callback disabled */
- #define USE_HAL_HASH_REGISTER_CALLBACKS 0U /* HASH register callback disabled */
- #define USE_HAL_HCD_REGISTER_CALLBACKS 0U /* HCD register callback disabled */
- #define USE_HAL_I2C_REGISTER_CALLBACKS 0U /* I2C register callback disabled */
- #define USE_HAL_I2S_REGISTER_CALLBACKS 0U /* I2S register callback disabled */
- #define USE_HAL_IRDA_REGISTER_CALLBACKS 0U /* IRDA register callback disabled */
- #define USE_HAL_JPEG_REGISTER_CALLBACKS 0U /* JPEG register callback disabled */
- #define USE_HAL_LPTIM_REGISTER_CALLBACKS 0U /* LPTIM register callback disabled */
- #define USE_HAL_LTDC_REGISTER_CALLBACKS 0U /* LTDC register callback disabled */
- #define USE_HAL_MDIOS_REGISTER_CALLBACKS 0U /* MDIOS register callback disabled */
- #define USE_HAL_MMC_REGISTER_CALLBACKS 0U /* MMC register callback disabled */
- #define USE_HAL_NAND_REGISTER_CALLBACKS 0U /* NAND register callback disabled */
- #define USE_HAL_NOR_REGISTER_CALLBACKS 0U /* NOR register callback disabled */
- #define USE_HAL_PCD_REGISTER_CALLBACKS 0U /* PCD register callback disabled */
- #define USE_HAL_QSPI_REGISTER_CALLBACKS 0U /* QSPI register callback disabled */
- #define USE_HAL_RNG_REGISTER_CALLBACKS 0U /* RNG register callback disabled */
- #define USE_HAL_RTC_REGISTER_CALLBACKS 0U /* RTC register callback disabled */
- #define USE_HAL_SAI_REGISTER_CALLBACKS 0U /* SAI register callback disabled */
- #define USE_HAL_SD_REGISTER_CALLBACKS 0U /* SD register callback disabled */
- #define USE_HAL_SMARTCARD_REGISTER_CALLBACKS 0U /* SMARTCARD register callback disabled */
- #define USE_HAL_SDRAM_REGISTER_CALLBACKS 0U /* SDRAM register callback disabled */
- #define USE_HAL_SRAM_REGISTER_CALLBACKS 0U /* SRAM register callback disabled */
- #define USE_HAL_SPDIFRX_REGISTER_CALLBACKS 0U /* SPDIFRX register callback disabled */
- #define USE_HAL_SMBUS_REGISTER_CALLBACKS 0U /* SMBUS register callback disabled */
- #define USE_HAL_SPI_REGISTER_CALLBACKS 0U /* SPI register callback disabled */
- #define USE_HAL_TIM_REGISTER_CALLBACKS 0U /* TIM register callback disabled */
- #define USE_HAL_UART_REGISTER_CALLBACKS 0U /* UART register callback disabled */
- #define USE_HAL_USART_REGISTER_CALLBACKS 0U /* USART register callback disabled */
- #define USE_HAL_WWDG_REGISTER_CALLBACKS 0U /* WWDG register callback disabled */
- #define MAC_ADDR0 2U

  Uncomment the line below to expanse the "assert_param" macro in the HAL drivers code.
- #define MAC_ADDR1 0U
- #define MAC_ADDR2 0U
- #define MAC_ADDR3 0U
- #define MAC_ADDR4 0U
- #define MAC_ADDR5 0U
- #define ETH_RX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for receive */
- #define ETH_TX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for transmit */
- #define ETH_RXBUFNB ((uint32_t)4U) /* 4 Rx buffers of size ETH_RX_BUF_SIZE */
- #define ETH_TXBUFNB ((uint32_t)4U) /* 4 Tx buffers of size ETH_TX_BUF_SIZE */
- #define DP83848_PHY_ADDRESS
- #define PHY_RESET_DELAY ((uint32_t)0x000000FFU)
- #define PHY_CONFIG_DELAY ((uint32_t)0x00000FFFU)
- #define PHY_READ_TO ((uint32_t)0x0000FFFFU)
- #define PHY_WRITE_TO ((uint32_t)0x0000FFFFU)
- #define PHY_BCR ((uint16_t)0x0000U)
- #define PHY_BSR ((uint16_t)0x0001U)
- #define PHY_RESET ((uint16_t)0x8000U)
- #define PHY_LOOPBACK ((uint16_t)0x4000U)
- #define PHY_FULLDUPLEX_100M ((uint16_t)0x2100U)
- #define PHY_HALFDUPLEX_100M ((uint16_t)0x2000U)

- #define PHY_FULLDUPLEX_10M ((uint16_t)0x0100U)
- #define PHY_HALFDUPLEX_10M ((uint16_t)0x0000U)
- #define PHY_AUTONEGOTIATION ((uint16_t)0x1000U)
- #define PHY_RESTART_AUTONEGOTIATION ((uint16_t)0x0200U)
- #define PHY_POWERDOWN ((uint16_t)0x0800U)
- #define PHY_ISOLATE ((uint16_t)0x0400U)
- #define PHY_AUTONEGO_COMPLETE ((uint16_t)0x0020U)
- #define PHY_LINKED_STATUS ((uint16_t)0x0004U)
- #define PHY_JABBER_DETECTION ((uint16_t)0x0002U)
- #define PHY_SR ((uint16_t))
- #define PHY_SPEED_STATUS ((uint16_t))
- #define PHY_DUPLEX_STATUS ((uint16_t))
- #define USE_SPI_CRC 0U
- #define assert_param(expr) ((void)0U)

Include module's header file.

6.23.1 Macro Definition Documentation

6.23.1.1 ART_ACCELERATOR_ENABLE

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

6.23.1.2 assert_param

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

Include module's header file.

6.23.1.3 DP83848_PHY_ADDRESS

```
#define DP83848_PHY_ADDRESS
```

6.23.1.4 ETH_RX_BUF_SIZE

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

6.23.1.5 ETH_RXBUFNB

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

6.23.1.6 ETH_TX_BUF_SIZE

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

6.23.1.7 ETH_TXBUFNB

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

6.23.1.8 EXTERNAL_CLOCK_VALUE

```
#define EXTERNAL_CLOCK_VALUE ((uint32_t)12288000U)
```

External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S_CKIN pad.

Value of the Internal oscillator in Hz

6.23.1.9 HAL_ADC_MODULE_ENABLED

```
#define HAL_ADC_MODULE_ENABLED
```

6.23.1.10 HAL_CAN_MODULE_ENABLED

```
#define HAL_CAN_MODULE_ENABLED
```

6.23.1.11 HAL_CORTEX_MODULE_ENABLED

```
#define HAL_CORTEX_MODULE_ENABLED
```

6.23.1.12 HAL_DAC_MODULE_ENABLED

```
#define HAL_DAC_MODULE_ENABLED
```

6.23.1.13 HAL_DMA_MODULE_ENABLED

```
#define HAL_DMA_MODULE_ENABLED
```

6.23.1.14 HAL_EXTI_MODULE_ENABLED

```
#define HAL_EXTI_MODULE_ENABLED
```

6.23.1.15 HAL_FLASH_MODULE_ENABLED

```
#define HAL_FLASH_MODULE_ENABLED
```

6.23.1.16 HAL_GPIO_MODULE_ENABLED

```
#define HAL_GPIO_MODULE_ENABLED
```

6.23.1.17 HAL_I2C_MODULE_ENABLED

```
#define HAL_I2C_MODULE_ENABLED
```

6.23.1.18 HAL_MODULE_ENABLED

```
#define HAL_MODULE_ENABLED
```

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

6.23.1.19 HAL_PWR_MODULE_ENABLED

```
#define HAL_PWR_MODULE_ENABLED
```

6.23.1.20 HAL_RCC_MODULE_ENABLED

```
#define HAL_RCC_MODULE_ENABLED
```

6.23.1.21 HAL_TIM_MODULE_ENABLED

```
#define HAL_TIM_MODULE_ENABLED
```

6.23.1.22 HSE_STARTUP_TIMEOUT

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

Time out for HSE start up, in ms

6.23.1.23 HSE_VALUE

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

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

Value of the External oscillator in Hz

6.23.1.24 HSI_VALUE

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

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

Value of the Internal oscillator in Hz

6.23.1.25 LSE_STARTUP_TIMEOUT

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

Time out for LSE start up, in ms

6.23.1.26 LSE_VALUE

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

External Low Speed oscillator (LSE) value.

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

6.23.1.27 LSI_VALUE

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

Internal Low Speed oscillator (LSI) value.

LSI Typical Value in Hz

6.23.1.28 MAC_ADDR0

```
#define MAC_ADDR0 2U
```

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

6.23.1.29 MAC_ADDR1

```
#define MAC_ADDR1 0U
```

6.23.1.30 MAC_ADDR2

```
#define MAC_ADDR2 0U
```

6.23.1.31 MAC_ADDR3

```
#define MAC_ADDR3 0U
```

6.23.1.32 MAC_ADDR4

```
#define MAC_ADDR4 0U
```

6.23.1.33 MAC_ADDR5

```
#define MAC_ADDR5 0U
```

6.23.1.34 PHY_AUTONEGO_COMPLETE

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

Auto-Negotiation process completed

6.23.1.35 PHY_AUTONEGOTIATION

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

Enable auto-negotiation function

6.23.1.36 PHY_BCR

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

Transceiver Basic Control Register

6.23.1.37 PHY_BSR

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

Transceiver Basic Status Register

6.23.1.38 PHY_CONFIG_DELAY

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

6.23.1.39 PHY_DUPLEX_STATUS

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

PHY Duplex mask

6.23.1.40 PHY_FULLDUPLEX_100M

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

Set the full-duplex mode at 100 Mb/s

6.23.1.41 PHY_FULLDUPLEX_10M

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

Set the full-duplex mode at 10 Mb/s

6.23.1.42 PHY_HALFDUPLEX_100M

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

Set the half-duplex mode at 100 Mb/s

6.23.1.43 PHY_HALFDUPLEX_10M

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

Set the half-duplex mode at 10 Mb/s

6.23.1.44 PHY_ISOLATE

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

Isolate PHY from MII

6.23.1.45 PHY_JABBER_DETECTION

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

Jabber condition detected

6.23.1.46 PHY_LINKED_STATUS

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

Valid link established

6.23.1.47 PHY_LOOPBACK

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

Select loop-back mode

6.23.1.48 PHY_POWERDOWN

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

Select the power down mode

6.23.1.49 PHY_READ_TO

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

6.23.1.50 PHY_RESET

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

PHY Reset

6.23.1.51 PHY_RESET_DELAY

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

6.23.1.52 PHY_RESTART_AUTONEGOTIATION

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

Restart auto-negotiation function

6.23.1.53 PHY_SPEED_STATUS

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

PHY Speed mask

6.23.1.54 PHY_SR

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

PHY status register Offset

6.23.1.55 PHY_WRITE_TO

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

6.23.1.56 PREFETCH_ENABLE

```
#define PREFETCH_ENABLE 0U
```

6.23.1.57 TICK_INT_PRIORITY

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

tick interrupt priority

6.23.1.58 USE_HAL_ADC_REGISTER_CALLBACKS

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

6.23.1.59 USE_HAL_CAN_REGISTER_CALLBACKS

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

6.23.1.60 USE_HAL_CEC_REGISTER_CALLBACKS

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

6.23.1.61 USE_HAL_CRYP_REGISTER_CALLBACKS

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

6.23.1.62 USE_HAL_DAC_REGISTER_CALLBACKS

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

6.23.1.63 USE_HAL_DCMI_REGISTER_CALLBACKS

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

6.23.1.64 USE_HAL_DFSDM_REGISTER_CALLBACKS

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

6.23.1.65 USE_HAL_DMA2D_REGISTER_CALLBACKS

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

6.23.1.66 USE_HAL_DSI_REGISTER_CALLBACKS

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

6.23.1.67 USE_HAL_ETH_REGISTER_CALLBACKS

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

6.23.1.68 USE_HAL_HASH_REGISTER_CALLBACKS

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

6.23.1.69 USE_HAL_HCD_REGISTER_CALLBACKS

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

6.23.1.70 USE_HAL_I2C_REGISTER_CALLBACKS

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

6.23.1.71 USE_HAL_I2S_REGISTER_CALLBACKS

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

6.23.1.72 USE_HAL_IRDA_REGISTER_CALLBACKS

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

6.23.1.73 USE_HAL_JPEG_REGISTER_CALLBACKS

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

6.23.1.74 USE_HAL_LPTIM_REGISTER_CALLBACKS

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

6.23.1.75 USE_HAL_LTDC_REGISTER_CALLBACKS

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

6.23.1.76 USE_HAL_MDIOS_REGISTER_CALLBACKS

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

6.23.1.77 USE_HAL_MMC_REGISTER_CALLBACKS

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

6.23.1.78 USE_HAL_NAND_REGISTER_CALLBACKS

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

6.23.1.79 USE_HAL_NOR_REGISTER_CALLBACKS

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

6.23.1.80 USE_HAL_PCD_REGISTER_CALLBACKS

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

6.23.1.81 USE_HAL_QSPI_REGISTER_CALLBACKS

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

6.23.1.82 USE_HAL_RNG_REGISTER_CALLBACKS

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

6.23.1.83 USE_HAL_RTC_REGISTER_CALLBACKS

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

6.23.1.84 USE_HAL_SAI_REGISTER_CALLBACKS

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

6.23.1.85 USE_HAL_SD_REGISTER_CALLBACKS

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

6.23.1.86 USE_HAL_SDRAM_REGISTER_CALLBACKS

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

6.23.1.87 USE_HAL_SMARTCARD_REGISTER_CALLBACKS

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

6.23.1.88 USE_HAL_SMBUS_REGISTER_CALLBACKS

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

6.23.1.89 USE_HAL_SPDIFRX_REGISTER_CALLBACKS

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

6.23.1.90 USE_HAL_SPI_REGISTER_CALLBACKS

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

6.23.1.91 USE_HAL_SRAM_REGISTER_CALLBACKS

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

6.23.1.92 USE_HAL_TIM_REGISTER_CALLBACKS

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

6.23.1.93 USE_HAL_UART_REGISTER_CALLBACKS

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

6.23.1.94 USE_HAL_USART_REGISTER_CALLBACKS

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

6.23.1.95 USE_HAL_WWDG_REGISTER_CALLBACKS

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

6.23.1.96 USE_RTOS

```
#define USE_RTOS 0U
```

6.23.1.97 USE_SPI_CRC

```
#define USE_SPI_CRC 0U
```

6.23.1.98 VDD_VALUE

```
#define VDD_VALUE 3300U
```

This is the HAL system configuration section.

Value of VDD in mv

6.24 stm32f7xx_hal_conf.h

Go to the documentation of this file.

```
00001 /* USER CODE BEGIN Header */
00021 /* USER CODE END Header */
00022
00023 /* Define to prevent recursive inclusion -------------------------------------*/
00024 #ifndef __STM32F7xx_HAL_CONF_H
00025 #define __STM32F7xx_HAL_CONF_H
00026
00027 #ifdef __cplusplus
00028  extern "C" {
00029 #endif
00030
00031 /* Exported types ------------------------------------------------------------*/
00032 /* Exported constants --------------------------------------------------------*/
00033
00034 /* ########################## Module Selection ############################## */
00038 #define HAL_MODULE_ENABLED
00039
00040   /* #define HAL_CRYP_MODULE_ENABLED */
00041 #define HAL_ADC_MODULE_ENABLED
00042 #define HAL_CAN_MODULE_ENABLED
00043 /* #define HAL_CEC_MODULE_ENABLED */
00044 /* #define HAL_CRC_MODULE_ENABLED */
00045 #define HAL_DAC_MODULE_ENABLED
```

```
00046 /* #define HAL_DCMI_MODULE_ENABLED */
00047 /* #define HAL_DMA2D_MODULE_ENABLED */
00048 /* #define HAL_ETH_MODULE_ENABLED */
00049 /* #define HAL_ETH_LEGACY_MODULE_ENABLED */
00050 /* #define HAL_NAND_MODULE_ENABLED */
00051 /* #define HAL_NOR_MODULE_ENABLED */
00052 /* #define HAL_SRAM_MODULE_ENABLED */
00053 /* #define HAL_SDRAM_MODULE_ENABLED */
00054 /* #define HAL_HASH_MODULE_ENABLED */
00055 /* #define HAL_I2S_MODULE_ENABLED */
00056 /* #define HAL_IWDG_MODULE_ENABLED */
00057 /* #define HAL_LPTIM_MODULE_ENABLED */
00058 /* #define HAL_LTDC_MODULE_ENABLED */
00059 /* #define HAL_QSPI_MODULE_ENABLED */
00060 /* #define HAL_RNG_MODULE_ENABLED */
00061 /* #define HAL_RTC_MODULE_ENABLED */
00062 /* #define HAL_SAI_MODULE_ENABLED */
00063 /* #define HAL_SD_MODULE_ENABLED */
00064 /* #define HAL_MMC_MODULE_ENABLED */
00065 /* #define HAL_SPDIFRX_MODULE_ENABLED */
00066 /* #define HAL_SPI_MODULE_ENABLED */
00067 #define HAL_TIM_MODULE_ENABLED
00068 /* #define HAL_UART_MODULE_ENABLED */
00069 /* #define HAL_USART_MODULE_ENABLED */
00070 /* #define HAL_IRDA_MODULE_ENABLED */
00071 /* #define HAL_SMARTCARD_MODULE_ENABLED */
00072 /* #define HAL_WWDG_MODULE_ENABLED */
00073 /* #define HAL_PCD_MODULE_ENABLED */
00074 /* #define HAL_HCD_MODULE_ENABLED */
00075 /* #define HAL_DFSDM_MODULE_ENABLED */
00076 /* #define HAL_DSI_MODULE_ENABLED */
00077 /* #define HAL_JPEG_MODULE_ENABLED */
00078 /* #define HAL_MDIOS_MODULE_ENABLED */
00079 /* #define HAL_SMBUS_MODULE_ENABLED */
00080 /* #define HAL_EXTI_MODULE_ENABLED */
00081 #define HAL_GPIO_MODULE_ENABLED
00082 #define HAL_EXTI_MODULE_ENABLED
00083 #define HAL_DMA_MODULE_ENABLED
00084 #define HAL_RCC_MODULE_ENABLED
00085 #define HAL_FLASH_MODULE_ENABLED
00086 #define HAL_PWR_MODULE_ENABLED
00087 #define HAL_I2C_MODULE_ENABLED
00088 #define HAL_CORTEX_MODULE_ENABLED
00089
00090 /* ########################## HSE/HSI Values adaptation ##################### */
00096 #if !defined  (HSE_VALUE)
00097   #define HSE_VALUE    ((uint32_t)20000000U)
00098 #endif /* HSE_VALUE */
00099
00100 #if !defined  (HSE_STARTUP_TIMEOUT)
00101   #define HSE_STARTUP_TIMEOUT    ((uint32_t)100U)
00102 #endif /* HSE_STARTUP_TIMEOUT */
00103
00109 #if !defined  (HSI_VALUE)
00110   #define HSI_VALUE    ((uint32_t)16000000U)
00111 #endif /* HSI_VALUE */
00112
00116 #if !defined  (LSI_VALUE)
00117  #define LSI_VALUE  ((uint32_t)32000U)
00118 #endif /* LSI_VALUE */
00124 #if !defined  (LSE_VALUE)
00125  #define LSE_VALUE  ((uint32_t)32768U)
00126 #endif /* LSE_VALUE */
00127
00128 #if !defined  (LSE_STARTUP_TIMEOUT)
00129   #define LSE_STARTUP_TIMEOUT    ((uint32_t)5000U)
00130 #endif /* LSE_STARTUP_TIMEOUT */
00131
00137 #if !defined  (EXTERNAL_CLOCK_VALUE)
00138   #define EXTERNAL_CLOCK_VALUE    ((uint32_t)12288000U)
00139 #endif /* EXTERNAL_CLOCK_VALUE */
00140
00141 /* Tip: To avoid modifying this file each time you need to use different HSE,
00142    ===  you can define the HSE value in your toolchain compiler preprocessor. */
00143
00144 /* ########################### System Configuration ######################### */
00148 #define  VDD_VALUE                    3300U
00149 #define  TICK_INT_PRIORITY            ((uint32_t)15U)
00150 #define  USE_RTOS                     0U
00151 #define  PREFETCH_ENABLE              0U
00152 #define  ART_ACCELERATOR_ENABLE        0U /* To enable instruction cache and prefetch */
00153
00154 #define  USE_HAL_ADC_REGISTER_CALLBACKS         0U /* ADC register callback disabled       */
00155 #define  USE_HAL_CAN_REGISTER_CALLBACKS         0U /* CAN register callback disabled       */
00156 #define  USE_HAL_CEC_REGISTER_CALLBACKS         0U /* CEC register callback disabled       */
00157 #define  USE_HAL_CRYP_REGISTER_CALLBACKS        0U /* CRYP register callback disabled      */
00158 #define  USE_HAL_DAC_REGISTER_CALLBACKS         0U /* DAC register callback disabled       */
```

```
00159 #define  USE_HAL_DCMI_REGISTER_CALLBACKS        0U /* DCMI register callback disabled      */
00160 #define  USE_HAL_DFSDM_REGISTER_CALLBACKS       0U /* DFSDM register callback disabled     */
00161 #define  USE_HAL_DMA2D_REGISTER_CALLBACKS       0U /* DMA2D register callback disabled     */
00162 #define  USE_HAL_DSI_REGISTER_CALLBACKS         0U /* DSI register callback disabled       */
00163 #define  USE_HAL_ETH_REGISTER_CALLBACKS         0U /* ETH register callback disabled       */
00164 #define  USE_HAL_HASH_REGISTER_CALLBACKS        0U /* HASH register callback disabled      */
00165 #define  USE_HAL_HCD_REGISTER_CALLBACKS         0U /* HCD register callback disabled       */
00166 #define  USE_HAL_I2C_REGISTER_CALLBACKS         0U /* I2C register callback disabled       */
00167 #define  USE_HAL_I2S_REGISTER_CALLBACKS         0U /* I2S register callback disabled       */
00168 #define  USE_HAL_IRDA_REGISTER_CALLBACKS        0U /* IRDA register callback disabled      */
00169 #define  USE_HAL_JPEG_REGISTER_CALLBACKS        0U /* JPEG register callback disabled      */
00170 #define  USE_HAL_LPTIM_REGISTER_CALLBACKS       0U /* LPTIM register callback disabled     */
00171 #define  USE_HAL_LTDC_REGISTER_CALLBACKS        0U /* LTDC register callback disabled      */
00172 #define  USE_HAL_MDIOS_REGISTER_CALLBACKS       0U /* MDIOS register callback disabled     */
00173 #define  USE_HAL_MMC_REGISTER_CALLBACKS         0U /* MMC register callback disabled       */
00174 #define  USE_HAL_NAND_REGISTER_CALLBACKS        0U /* NAND register callback disabled      */
00175 #define  USE_HAL_NOR_REGISTER_CALLBACKS         0U /* NOR register callback disabled       */
00176 #define  USE_HAL_PCD_REGISTER_CALLBACKS         0U /* PCD register callback disabled       */
00177 #define  USE_HAL_QSPI_REGISTER_CALLBACKS        0U /* QSPI register callback disabled      */
00178 #define  USE_HAL_RNG_REGISTER_CALLBACKS         0U /* RNG register callback disabled       */
00179 #define  USE_HAL_RTC_REGISTER_CALLBACKS         0U /* RTC register callback disabled       */
00180 #define  USE_HAL_SAI_REGISTER_CALLBACKS         0U /* SAI register callback disabled       */
00181 #define  USE_HAL_SD_REGISTER_CALLBACKS          0U /* SD register callback disabled        */
00182 #define  USE_HAL_SMARTCARD_REGISTER_CALLBACKS   0U /* SMARTCARD register callback disabled */
00183 #define  USE_HAL_SDRAM_REGISTER_CALLBACKS       0U /* SDRAM register callback disabled     */
00184 #define  USE_HAL_SRAM_REGISTER_CALLBACKS        0U /* SRAM register callback disabled      */
00185 #define  USE_HAL_SPDIFRX_REGISTER_CALLBACKS     0U /* SPDIFRX register callback disabled   */
00186 #define  USE_HAL_SMBUS_REGISTER_CALLBACKS       0U /* SMBUS register callback disabled     */
00187 #define  USE_HAL_SPI_REGISTER_CALLBACKS         0U /* SPI register callback disabled       */
00188 #define  USE_HAL_TIM_REGISTER_CALLBACKS         0U /* TIM register callback disabled       */
00189 #define  USE_HAL_UART_REGISTER_CALLBACKS        0U /* UART register callback disabled      */
00190 #define  USE_HAL_USART_REGISTER_CALLBACKS       0U /* USART register callback disabled     */
00191 #define  USE_HAL_WWDG_REGISTER_CALLBACKS        0U /* WWDG register callback disabled      */
00192
00193 /* ########################## Assert Selection ############################## */
00198 /* #define USE_FULL_ASSERT    1U */
00199
00200 /* ################### Ethernet peripheral configuration ##################### */
00201
00202 /* Section 1 : Ethernet peripheral configuration */
00203
00204 /* MAC ADDRESS: MAC_ADDR0:MAC_ADDR1:MAC_ADDR2:MAC_ADDR3:MAC_ADDR4:MAC_ADDR5 */
00205 #define MAC_ADDR0   2U
00206 #define MAC_ADDR1   0U
00207 #define MAC_ADDR2   0U
00208 #define MAC_ADDR3   0U
00209 #define MAC_ADDR4   0U
00210 #define MAC_ADDR5   0U
00211
00212 /* Definition of the Ethernet driver buffers size and count */
00213 #define ETH_RX_BUF_SIZE                ETH_MAX_PACKET_SIZE /* buffer size for receive               */
00214 #define ETH_TX_BUF_SIZE                ETH_MAX_PACKET_SIZE /* buffer size for transmit              */
00215 #define ETH_RXBUFNB                    ((uint32_t)4U)       /* 4 Rx buffers of size ETH_RX_BUF_SIZE
      */
00216 #define ETH_TXBUFNB                    ((uint32_t)4U)       /* 4 Tx buffers of size ETH_TX_BUF_SIZE
      */
00217
00218 /* Section 2: PHY configuration section */
00219
00220 /* DP83848_PHY_ADDRESS Address*/
00221 #define DP83848_PHY_ADDRESS
00222 /* PHY Reset delay these values are based on a 1 ms Systick interrupt*/
00223 #define PHY_RESET_DELAY                 ((uint32_t)0x000000FFU)
00224 /* PHY Configuration delay */
00225 #define PHY_CONFIG_DELAY                ((uint32_t)0x00000FFFU)
00226
00227 #define PHY_READ_TO                     ((uint32_t)0x0000FFFFU)
00228 #define PHY_WRITE_TO                    ((uint32_t)0x0000FFFFU)
00229
00230 /* Section 3: Common PHY Registers */
00231
00232 #define PHY_BCR                         ((uint16_t)0x0000U)
00233 #define PHY_BSR                         ((uint16_t)0x0001U)
00235 #define PHY_RESET                       ((uint16_t)0x8000U)
00236 #define PHY_LOOPBACK                    ((uint16_t)0x4000U)
00237 #define PHY_FULLDUPLEX_100M             ((uint16_t)0x2100U)
00238 #define PHY_HALFDUPLEX_100M             ((uint16_t)0x2000U)
00239 #define PHY_FULLDUPLEX_10M              ((uint16_t)0x0100U)
00240 #define PHY_HALFDUPLEX_10M              ((uint16_t)0x0000U)
00241 #define PHY_AUTONEGOTIATION             ((uint16_t)0x1000U)
00242 #define PHY_RESTART_AUTONEGOTIATION     ((uint16_t)0x0200U)
00243 #define PHY_POWERDOWN                   ((uint16_t)0x0800U)
00244 #define PHY_ISOLATE                     ((uint16_t)0x0400U)
00246 #define PHY_AUTONEGO_COMPLETE           ((uint16_t)0x0020U)
00247 #define PHY_LINKED_STATUS               ((uint16_t)0x0004U)
00248 #define PHY_JABBER_DETECTION            ((uint16_t)0x0002U)
00250 /* Section 4: Extended PHY Registers */
```

```
00251 #define PHY_SR                           ((uint16_t))
00253 #define PHY_SPEED_STATUS                 ((uint16_t))
00254 #define PHY_DUPLEX_STATUS                ((uint16_t))
00256 /* ################## SPI peripheral configuration ########################## */
00257
00258 /* CRC FEATURE: Use to activate CRC feature inside HAL SPI Driver
00259 * Activated: CRC code is present inside driver
00260 * Deactivated: CRC code cleaned from driver
00261 */
00262
00263 #define USE_SPI_CRC                      0U
00264
00265 /* Includes ------------------------------------------------------------------*/
00270 #ifdef HAL_RCC_MODULE_ENABLED
00271   #include "stm32f7xx_hal_rcc.h"
00272 #endif /* HAL_RCC_MODULE_ENABLED */
00273
00274 #ifdef HAL_EXTI_MODULE_ENABLED
00275   #include "stm32f7xx_hal_exti.h"
00276 #endif /* HAL_EXTI_MODULE_ENABLED */
00277
00278 #ifdef HAL_GPIO_MODULE_ENABLED
00279   #include "stm32f7xx_hal_gpio.h"
00280 #endif /* HAL_GPIO_MODULE_ENABLED */
00281
00282 #ifdef HAL_DMA_MODULE_ENABLED
00283   #include "stm32f7xx_hal_dma.h"
00284 #endif /* HAL_DMA_MODULE_ENABLED */
00285
00286 #ifdef HAL_CORTEX_MODULE_ENABLED
00287   #include "stm32f7xx_hal_cortex.h"
00288 #endif /* HAL_CORTEX_MODULE_ENABLED */
00289
00290 #ifdef HAL_ADC_MODULE_ENABLED
00291   #include "stm32f7xx_hal_adc.h"
00292 #endif /* HAL_ADC_MODULE_ENABLED */
00293
00294 #ifdef HAL_CAN_MODULE_ENABLED
00295   #include "stm32f7xx_hal_can.h"
00296 #endif /* HAL_CAN_MODULE_ENABLED */
00297
00298 #ifdef HAL_CEC_MODULE_ENABLED
00299   #include "stm32f7xx_hal_cec.h"
00300 #endif /* HAL_CEC_MODULE_ENABLED */
00301
00302 #ifdef HAL_CRC_MODULE_ENABLED
00303   #include "stm32f7xx_hal_crc.h"
00304 #endif /* HAL_CRC_MODULE_ENABLED */
00305
00306 #ifdef HAL_CRYP_MODULE_ENABLED
00307   #include "stm32f7xx_hal_cryp.h"
00308 #endif /* HAL_CRYP_MODULE_ENABLED */
00309
00310 #ifdef HAL_DMA2D_MODULE_ENABLED
00311   #include "stm32f7xx_hal_dma2d.h"
00312 #endif /* HAL_DMA2D_MODULE_ENABLED */
00313
00314 #ifdef HAL_DAC_MODULE_ENABLED
00315   #include "stm32f7xx_hal_dac.h"
00316 #endif /* HAL_DAC_MODULE_ENABLED */
00317
00318 #ifdef HAL_DCMI_MODULE_ENABLED
00319   #include "stm32f7xx_hal_dcmi.h"
00320 #endif /* HAL_DCMI_MODULE_ENABLED */
00321
00322 #ifdef HAL_ETH_MODULE_ENABLED
00323   #include "stm32f7xx_hal_eth.h"
00324 #endif /* HAL_ETH_MODULE_ENABLED */
00325
00326 #ifdef HAL_ETH_LEGACY_MODULE_ENABLED
00327   #include "stm32f7xx_hal_eth_legacy.h"
00328 #endif /* HAL_ETH_LEGACY_MODULE_ENABLED */
00329
00330 #ifdef HAL_FLASH_MODULE_ENABLED
00331   #include "stm32f7xx_hal_flash.h"
00332 #endif /* HAL_FLASH_MODULE_ENABLED */
00333
00334 #ifdef HAL_SRAM_MODULE_ENABLED
00335   #include "stm32f7xx_hal_sram.h"
00336 #endif /* HAL_SRAM_MODULE_ENABLED */
00337
00338 #ifdef HAL_NOR_MODULE_ENABLED
00339   #include "stm32f7xx_hal_nor.h"
00340 #endif /* HAL_NOR_MODULE_ENABLED */
00341
00342 #ifdef HAL_NAND_MODULE_ENABLED
00343   #include "stm32f7xx_hal_nand.h"
```

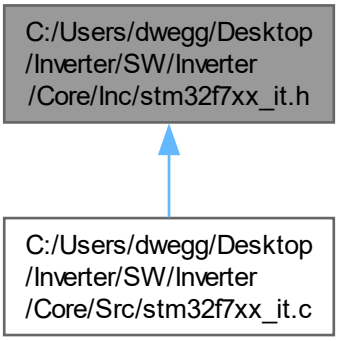
```
00344 #endif /* HAL_NAND_MODULE_ENABLED */
00345
00346 #ifdef HAL_SDRAM_MODULE_ENABLED
00347   #include "stm32f7xx_hal_sdram.h"
00348 #endif /* HAL_SDRAM_MODULE_ENABLED */
00349
00350 #ifdef HAL_HASH_MODULE_ENABLED
00351  #include "stm32f7xx_hal_hash.h"
00352 #endif /* HAL_HASH_MODULE_ENABLED */
00353
00354 #ifdef HAL_I2C_MODULE_ENABLED
00355  #include "stm32f7xx_hal_i2c.h"
00356 #endif /* HAL_I2C_MODULE_ENABLED */
00357
00358 #ifdef HAL_I2S_MODULE_ENABLED
00359  #include "stm32f7xx_hal_i2s.h"
00360 #endif /* HAL_I2S_MODULE_ENABLED */
00361
00362 #ifdef HAL_IWDG_MODULE_ENABLED
00363  #include "stm32f7xx_hal_iwdg.h"
00364 #endif /* HAL_IWDG_MODULE_ENABLED */
00365
00366 #ifdef HAL_LPTIM_MODULE_ENABLED
00367  #include "stm32f7xx_hal_lptim.h"
00368 #endif /* HAL_LPTIM_MODULE_ENABLED */
00369
00370 #ifdef HAL_LTDC_MODULE_ENABLED
00371  #include "stm32f7xx_hal_ltdc.h"
00372 #endif /* HAL_LTDC_MODULE_ENABLED */
00373
00374 #ifdef HAL_PWR_MODULE_ENABLED
00375  #include "stm32f7xx_hal_pwr.h"
00376 #endif /* HAL_PWR_MODULE_ENABLED */
00377
00378 #ifdef HAL_QSPI_MODULE_ENABLED
00379  #include "stm32f7xx_hal_qspi.h"
00380 #endif /* HAL_QSPI_MODULE_ENABLED */
00381
00382 #ifdef HAL_RNG_MODULE_ENABLED
00383  #include "stm32f7xx_hal_rng.h"
00384 #endif /* HAL_RNG_MODULE_ENABLED */
00385
00386 #ifdef HAL_RTC_MODULE_ENABLED
00387  #include "stm32f7xx_hal_rtc.h"
00388 #endif /* HAL_RTC_MODULE_ENABLED */
00389
00390 #ifdef HAL_SAI_MODULE_ENABLED
00391  #include "stm32f7xx_hal_sai.h"
00392 #endif /* HAL_SAI_MODULE_ENABLED */
00393
00394 #ifdef HAL_SD_MODULE_ENABLED
00395  #include "stm32f7xx_hal_sd.h"
00396 #endif /* HAL_SD_MODULE_ENABLED */
00397
00398 #ifdef HAL_MMC_MODULE_ENABLED
00399  #include "stm32f7xx_hal_mmc.h"
00400 #endif /* HAL_MMC_MODULE_ENABLED */
00401
00402 #ifdef HAL_SPDIFRX_MODULE_ENABLED
00403  #include "stm32f7xx_hal_spdifrx.h"
00404 #endif /* HAL_SPDIFRX_MODULE_ENABLED */
00405
00406 #ifdef HAL_SPI_MODULE_ENABLED
00407  #include "stm32f7xx_hal_spi.h"
00408 #endif /* HAL_SPI_MODULE_ENABLED */
00409
00410 #ifdef HAL_TIM_MODULE_ENABLED
00411  #include "stm32f7xx_hal_tim.h"
00412 #endif /* HAL_TIM_MODULE_ENABLED */
00413
00414 #ifdef HAL_UART_MODULE_ENABLED
00415  #include "stm32f7xx_hal_uart.h"
00416 #endif /* HAL_UART_MODULE_ENABLED */
00417
00418 #ifdef HAL_USART_MODULE_ENABLED
00419  #include "stm32f7xx_hal_usart.h"
00420 #endif /* HAL_USART_MODULE_ENABLED */
00421
00422 #ifdef HAL_IRDA_MODULE_ENABLED
00423  #include "stm32f7xx_hal_irda.h"
00424 #endif /* HAL_IRDA_MODULE_ENABLED */
00425
00426 #ifdef HAL_SMARTCARD_MODULE_ENABLED
00427  #include "stm32f7xx_hal_smartcard.h"
00428 #endif /* HAL_SMARTCARD_MODULE_ENABLED */
00429
00430 #ifdef HAL_WWDG_MODULE_ENABLED
```

```
00431  #include "stm32f7xx_hal_wwdg.h"
00432 #endif /* HAL_WWDG_MODULE_ENABLED */
00433
00434 #ifdef HAL_PCD_MODULE_ENABLED
00435  #include "stm32f7xx_hal_pcd.h"
00436 #endif /* HAL_PCD_MODULE_ENABLED */
00437
00438 #ifdef HAL_HCD_MODULE_ENABLED
00439  #include "stm32f7xx_hal_hcd.h"
00440 #endif /* HAL_HCD_MODULE_ENABLED */
00441
00442 #ifdef HAL_DFSDM_MODULE_ENABLED
00443  #include "stm32f7xx_hal_dfsdm.h"
00444 #endif /* HAL_DFSDM_MODULE_ENABLED */
00445
00446 #ifdef HAL_DSI_MODULE_ENABLED
00447  #include "stm32f7xx_hal_dsi.h"
00448 #endif /* HAL_DSI_MODULE_ENABLED */
00449
00450 #ifdef HAL_JPEG_MODULE_ENABLED
00451  #include "stm32f7xx_hal_jpeg.h"
00452 #endif /* HAL_JPEG_MODULE_ENABLED */
00453
00454 #ifdef HAL_MDIOS_MODULE_ENABLED
00455  #include "stm32f7xx_hal_mdios.h"
00456 #endif /* HAL_MDIOS_MODULE_ENABLED */
00457
00458 #ifdef HAL_SMBUS_MODULE_ENABLED
00459  #include "stm32f7xx_hal_smbus.h"
00460 #endif /* HAL_SMBUS_MODULE_ENABLED */
00461
00462 /* Exported macro ------------------------------------------------------------*/
00463 #ifdef  USE_FULL_ASSERT
00472   #define assert_param(expr) ((expr) ? (void)0U : assert_failed((uint8_t *)__FILE__, __LINE__))
00473 /* Exported functions ------------------------------------------------------- */
00474   void assert_failed(uint8_t* file, uint32_t line);
00475 #else
00476   #define assert_param(expr) ((void)0U)
00477 #endif /* USE_FULL_ASSERT */
00478
00479 #ifdef __cplusplus
00480 }
00481 #endif
00482
00483 #endif /* __STM32F7xx_HAL_CONF_H */
00484
```

6.25 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/stm32f7xx_↩ it.h File Reference

This file contains the headers of the interrupt handlers.

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



Functions

- void NMI_Handler (void)

  This function handles Non maskable interrupt.
- void HardFault_Handler (void)

  This function handles Hard fault interrupt.
- void MemManage_Handler (void)

  This function handles Memory management fault.
- void BusFault_Handler (void)

  This function handles Pre-fetch fault, memory access fault.
- void UsageFault_Handler (void)

  This function handles Undefined instruction or illegal state.
- void SVC_Handler (void)

  This function handles System service call via SWI instruction.
- void DebugMon_Handler (void)

  This function handles Debug monitor.
- void PendSV_Handler (void)

  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 CAN1_RX1_IRQHandler (void)

  This function handles CAN1 RX1 interrupt.
- void TIM1_BRK_TIM9_IRQHandler (void)

  This function handles TIM1 break interrupt and TIM9 global interrupt.
- void TIM1_UP_TIM10_IRQHandler (void)

  This function handles TIM1 update interrupt and TIM10 global interrupt.
- void TIM1_TRG_COM_TIM11_IRQHandler (void)

  This function handles TIM1 trigger and commutation interrupts and TIM11 global interrupt.
- void TIM1_CC_IRQHandler (void)

  This function handles TIM1 capture compare 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.

6.25.1 Detailed Description

This file contains the headers of the interrupt handlers.

Attention

6.25.2 Function Documentation

6.25.2.1 BusFault_Handler()

```
void BusFault_Handler (
            void  )
```

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

6.25.2.2 CAN1_RX0_IRQHandler()

```
void CAN1_RX0_IRQHandler (
            void  )
```

This function handles CAN1 RX0 interrupts.

6.25.2.3 CAN1_RX1_IRQHandler()

```
void CAN1_RX1_IRQHandler (
            void  )
```

This function handles CAN1 RX1 interrupt.

6.25.2.4 DebugMon_Handler()

```
void DebugMon_Handler (
            void  )
```

This function handles Debug monitor.

6.25.2.5 DMA2_Stream0_IRQHandler()

```
void DMA2_Stream0_IRQHandler (
            void  )
```

This function handles DMA2 stream0 global interrupt.

6.25.2.6 DMA2_Stream1_IRQHandler()

```
void DMA2_Stream1_IRQHandler (
            void  )
```

This function handles DMA2 stream1 global interrupt.

6.25.2.7 DMA2_Stream2_IRQHandler()

```
void DMA2_Stream2_IRQHandler (
             void  )
```

This function handles DMA2 stream2 global interrupt.

6.25.2.8 HardFault_Handler()

```
void HardFault_Handler (
             void  )
```

This function handles Hard fault interrupt.

6.25.2.9 MemManage_Handler()

```
void MemManage_Handler (
             void  )
```

This function handles Memory management fault.

6.25.2.10 NMI_Handler()

```
void NMI_Handler (
             void  )
```

This function handles Non maskable interrupt.

6.25.2.11 PendSV_Handler()

```
void PendSV_Handler (
             void  )
```

This function handles Pendable request for system service.

6.25.2.12 SVC_Handler()

```
void SVC_Handler (
             void  )
```

This function handles System service call via SWI instruction.

6.25.2.13 SysTick_Handler()

```
void SysTick_Handler (
             void  )
```

This function handles System tick timer.

6.25.2.14 TIM1_BRK_TIM9_IRQHandler()

```
void TIM1_BRK_TIM9_IRQHandler (
            void  )
```

This function handles TIM1 break interrupt and TIM9 global interrupt.

6.25.2.15 TIM1_CC_IRQHandler()

```
void TIM1_CC_IRQHandler (
            void  )
```

This function handles TIM1 capture compare interrupt.

6.25.2.16 TIM1_TRG_COM_TIM11_IRQHandler()

```
void TIM1_TRG_COM_TIM11_IRQHandler (
            void  )
```

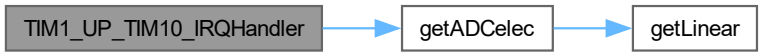
This function handles TIM1 trigger and commutation interrupts and TIM11 global interrupt.

6.25.2.17 TIM1_UP_TIM10_IRQHandler()

```
void TIM1_UP_TIM10_IRQHandler (
            void  )
```

This function handles TIM1 update interrupt and TIM10 global interrupt.

Here is the call graph for this function:

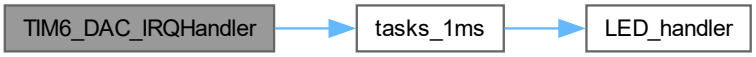


6.25.2.18 TIM6_DAC_IRQHandler()

```
void TIM6_DAC_IRQHandler (
            void  )
```

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

Here is the call graph for this function:



6.25.2.19 UsageFault_Handler()

```
void UsageFault_Handler (
             void  )
```

This function handles Undefined instruction or illegal state.

6.26 stm32f7xx_it.h

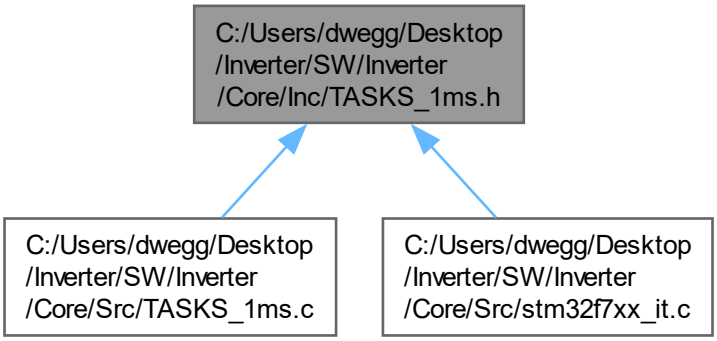
Go to the documentation of this file.

```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __STM32F7xx_IT_H
00022 #define __STM32F7xx_IT_H
00023
00024 #ifdef __cplusplus
00025  extern "C" {
00026 #endif
00027
00028 /* Private includes ----------------------------------------------------------*/
00029 /* USER CODE BEGIN Includes */
00030
00031 /* USER CODE END Includes */
00032
00033 /* Exported types ------------------------------------------------------------*/
00034 /* USER CODE BEGIN ET */
00035
00036 /* USER CODE END ET */
00037
00038 /* Exported constants --------------------------------------------------------*/
00039 /* USER CODE BEGIN EC */
00040
00041 /* USER CODE END EC */
00042
00043 /* Exported macro ------------------------------------------------------------*/
00044 /* USER CODE BEGIN EM */
00045
00046 /* USER CODE END EM */
00047
00048 /* Exported functions prototypes ---------------------------------------------*/
00049 void NMI_Handler(void);
00050 void HardFault_Handler(void);
00051 void MemManage_Handler(void);
00052 void BusFault_Handler(void);
00053 void UsageFault_Handler(void);
00054 void SVC_Handler(void);
00055 void DebugMon_Handler(void);
00056 void PendSV_Handler(void);
00057 void SysTick_Handler(void);
00058 void CAN1_RX0_IRQHandler(void);
00059 void CAN1_RX1_IRQHandler(void);
00060 void TIM1_BRK_TIM9_IRQHandler(void);
00061 void TIM1_UP_TIM10_IRQHandler(void);
00062 void TIM1_TRG_COM_TIM11_IRQHandler(void);
00063 void TIM1_CC_IRQHandler(void);
00064 void TIM6_DAC_IRQHandler(void);
00065 void DMA2_Stream0_IRQHandler(void);
00066 void DMA2_Stream1_IRQHandler(void);
00067 void DMA2_Stream2_IRQHandler(void);
00068 /* USER CODE BEGIN EFP */
00069
00070 /* USER CODE END EFP */
00071
00072 #ifdef __cplusplus
00073 }
00074 #endif
00075
00076 #endif /* __STM32F7xx_IT_H */
```

6.27 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/TASKS_1ms.h File Reference

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

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



Functions

- void tasks_1ms (void)

  Function to be executed every 1ms.

6.27.1 Detailed Description

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

Attention

6.27.2 Function Documentation

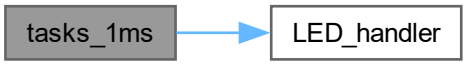
6.27.2.1 tasks_1ms()

```
void tasks_1ms (
            void  )
```

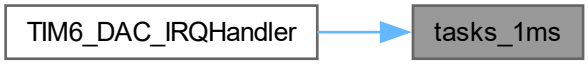
Function to be executed every 1ms.

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

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



Here is the caller graph for this function:



6.28 TASKS_1ms.h

Go to the documentation of this file.

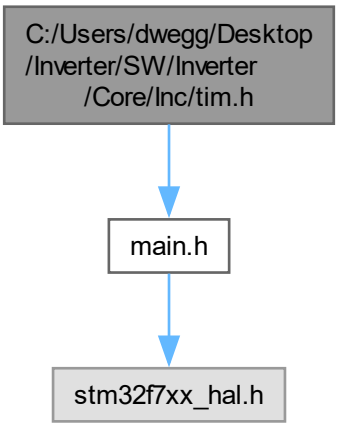
```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020
00021 #ifndef TASKS_1MS_H
00022 #define TASKS_1MS_H
00023
00029 void tasks_1ms(void);
00030
00031 #endif /* TASKS_1MS_H */
```

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

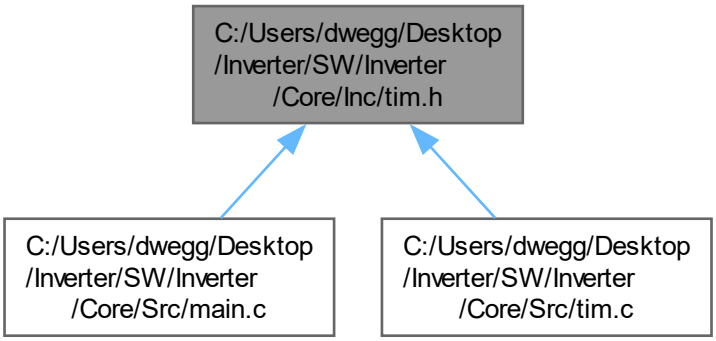
This file contains all the function prototypes for the tim.c file.

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

Include dependency graph for tim.h:



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



Functions

- void MX_TIM1_Init (void)
- void MX_TIM2_Init (void)
- void MX_TIM4_Init (void)
- void MX_TIM6_Init (void)
- void MX_TIM8_Init (void)
- void HAL_TIM_MspPostInit (TIM_HandleTypeDef *htim)

Variables

- TIM_HandleTypeDef htim1
- TIM_HandleTypeDef htim2
- TIM_HandleTypeDef htim4
- TIM_HandleTypeDef htim6
- TIM_HandleTypeDef htim8

6.29.1 Detailed Description

This file contains all the function prototypes for the tim.c file.

Attention

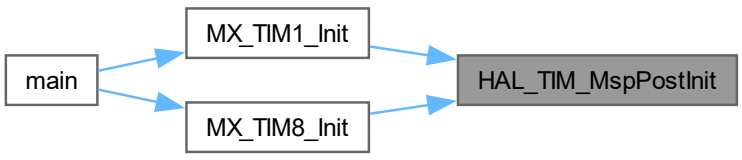
6.29.2 Function Documentation

6.29.2.1 HAL_TIM_MspPostInit()

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

TIM1 GPIO Configuration PE8 ------> TIM1_CH1N PE9 ------> TIM1_CH1 PE10 ------> TIM1_CH2N PE11 ------> TIM1_CH2 PE12 ------> TIM1_CH3N PE13 ------> TIM1_CH3

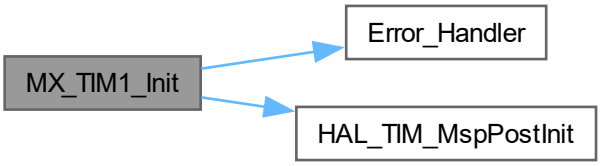
TIM8 GPIO Configuration PA5 ------> TIM8_CH1N PB14 ------> TIM8_CH2N PB15 ------> TIM8_CH3N PC6 ------> TIM8_CH1 PC7 ------> TIM8_CH2 PC8 ------> TIM8_CH3Here is the caller graph for this function:



6.29.2.2 MX_TIM1_Init()

```
void MX_TIM1_Init (
            void  )
```

Here is the call graph for this function:

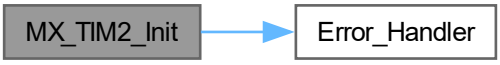


Here is the caller graph for this function:

6.29.2.3 MX_TIM2_Init()

```
void MX_TIM2_Init (
            void  )
```

Here is the call graph for this function:

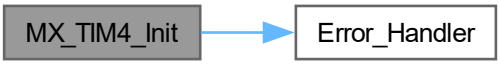


Here is the caller graph for this function:

6.29.2.4 MX_TIM4_Init()

```
void MX_TIM4_Init (
              void  )
```

Here is the call graph for this function:

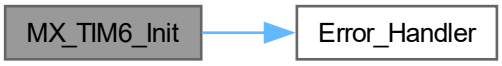


Here is the caller graph for this function:

6.29.2.5 MX_TIM6_Init()

```
void MX_TIM6_Init (
              void  )
```

Here is the call graph for this function:

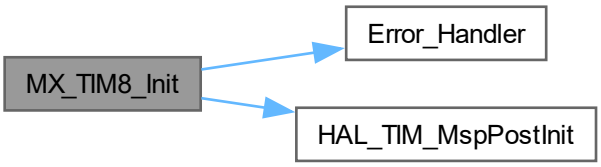


Here is the caller graph for this function:

6.29.2.6 MX_TIM8_Init()

```
void MX_TIM8_Init (
             void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:

6.29.3 Variable Documentation

6.29.3.1 htim1

```
TIM_HandleTypeDef htim1  [extern]
```

6.29.3.2 htim2

```
TIM_HandleTypeDef htim2  [extern]
```

6.29.3.3 htim4

```
TIM_HandleTypeDef htim4  [extern]
```

6.29.3.4 htim6

```
TIM_HandleTypeDef htim6  [extern]
```

6.29.3.5 htim8

```
TIM_HandleTypeDef htim8  [extern]
```

6.30 tim.h

Go to the documentation of this file.

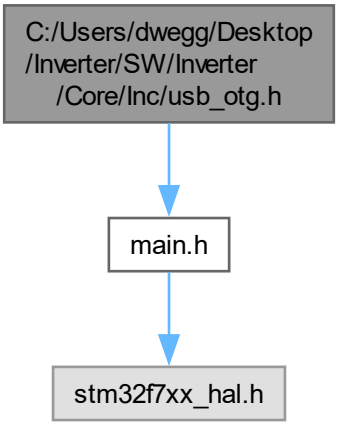
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __TIM_H__
00022 #define __TIM_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes ------------------------------------------------------------------*/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

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

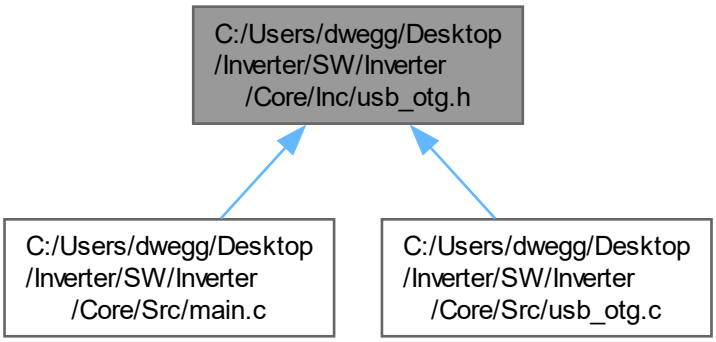
6.31 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Inc/usb_otg.h File Reference

This file contains all the function prototypes for the usb_otg.c file.

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



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



Functions

- void MX_USB_OTG_FS_USB_Init (void)

6.31.1 Detailed Description

This file contains all the function prototypes for the usb_otg.c file.

Attention

6.31.2 Function Documentation

6.31.2.1 MX_USB_OTG_FS_USB_Init()

```
void MX_USB_OTG_FS_USB_Init (
            void  )
```

Here is the caller graph for this function:

6.32 usb_otg.h

Go to the documentation of this file.

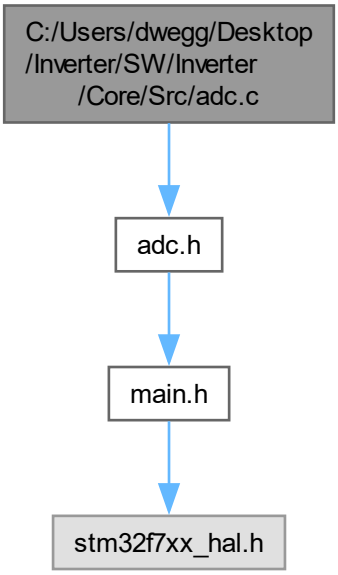
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __USB_OTG_H__
00022 #define __USB_OTG_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes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

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

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

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

Include dependency graph for adc.c:



Functions

- void MX_ADC1_Init (void)
- void MX_ADC2_Init (void)
- void MX_ADC3_Init (void)
- void HAL_ADC_MspInit (ADC_HandleTypeDef *adcHandle)
- void HAL_ADC_MspDeInit (ADC_HandleTypeDef *adcHandle)

Variables

- ADC_HandleTypeDef hadc1
- ADC_HandleTypeDef hadc2
- ADC_HandleTypeDef hadc3
- DMA_HandleTypeDef hdma_adc1
- DMA_HandleTypeDef hdma_adc2
- DMA_HandleTypeDef hdma_adc3

6.33.1 Detailed Description

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

Attention

6.33.2 Function Documentation

6.33.2.1 HAL_ADC_MspDeInit()

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

ADC1 GPIO Configuration PA0/WKUP ------> ADC1_IN0 PA1 ------> ADC1_IN1 PA2 ------> ADC1_IN2 PA3 ------> ADC1_IN3

ADC2 GPIO Configuration PA6 ------> ADC2_IN6 PA7 ------> ADC2_IN7 PB0 ------> ADC2_IN8 PB1 ------> ADC2_IN9

ADC3 GPIO Configuration PC0 ------> ADC3_IN10 PC1 ------> ADC3_IN11 PC2 ------> ADC3_IN12 PC3 ------> ADC3_IN13

6.33.2.2 HAL_ADC_MspInit()

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

ADC1 GPIO Configuration PA0/WKUP ------> ADC1_IN0 PA1 ------> ADC1_IN1 PA2 ------> ADC1_IN2 PA3 ------> ADC1_IN3

ADC2 GPIO Configuration PA6 ------> ADC2_IN6 PA7 ------> ADC2_IN7 PB0 ------> ADC2_IN8 PB1 ------> ADC2_IN9

ADC3 GPIO Configuration PC0 ------> ADC3_IN10 PC1 ------> ADC3_IN11 PC2 ------> ADC3_IN12 PC3 ------> ADC3_IN13Here is the call graph for this function:

6.33.2.3 MX_ADC1_Init()

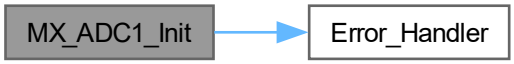
```
void MX_ADC1_Init (
            void  )
```

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

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

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

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



Here is the caller graph for this function:

6.33.2.4 MX_ADC2_Init()

```
void MX_ADC2_Init (
             void  )
```

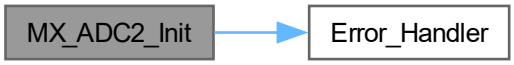
Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of conversion)

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

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

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

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



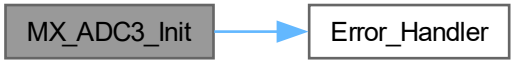
Here is the caller graph for this function:

6.33.2.5 MX_ADC3_Init()

```
void MX_ADC3_Init (
             void  )
```

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

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



Here is the caller graph for this function:

6.33.3 Variable Documentation

6.33.3.1 hadc1

```
ADC_HandleTypeDef hadc1
```

6.33.3.2 hadc2

```
ADC_HandleTypeDef hadc2
```

6.33.3.3 hadc3

```
ADC_HandleTypeDef hadc3
```

6.33.3.4 hdma_adc1

```
DMA_HandleTypeDef hdma_adc1
```

6.33.3.5 hdma_adc2

```
DMA_HandleTypeDef hdma_adc2
```

6.33.3.6 hdma_adc3

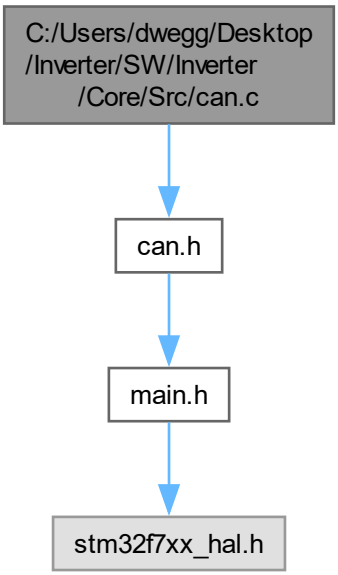
```
DMA_HandleTypeDef hdma_adc3
```

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

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

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

Include dependency graph for can.c:



Functions

- void MX_CAN1_Init (void)
- void HAL_CAN_MspInit (CAN_HandleTypeDef ∗canHandle)
- void HAL_CAN_MspDeInit (CAN_HandleTypeDef ∗canHandle)

Variables

- CAN_HandleTypeDef hcan1

6.34.1 Detailed Description

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

Attention

6.34.2 Function Documentation

6.34.2.1 HAL_CAN_MspDeInit()

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

CAN1 GPIO Configuration PD0 ------> CAN1_RX PD1 ------> CAN1_TX

6.34.2.2 HAL_CAN_MspInit()

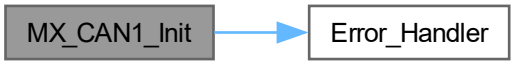
```
void HAL_CAN_MspInit (
            CAN_HandleTypeDef * canHandle )
```

CAN1 GPIO Configuration PD0 ------> CAN1_RX PD1 ------> CAN1_TX

6.34.2.3 MX_CAN1_Init()

```
void MX_CAN1_Init (
             void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:

6.34.3 Variable Documentation

6.34.3.1 hcan1

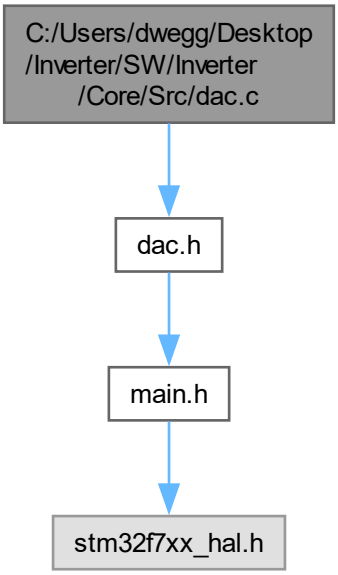
```
CAN_HandleTypeDef hcan1
```

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

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

```
#include "dac.h"
```

Include dependency graph for dac.c:



Functions

- void MX_DAC_Init (void)
- void HAL_DAC_MspInit (DAC_HandleTypeDef *dacHandle)
- void HAL_DAC_MspDeInit (DAC_HandleTypeDef *dacHandle)

Variables

- DAC_HandleTypeDef hdac

6.35.1 Detailed Description

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

Attention

6.35.2 Function Documentation

6.35.2.1 HAL_DAC_MspDeInit()

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

DAC GPIO Configuration PA4 ------> DAC_OUT1

Uncomment the line below to disable the "TIM6_DAC_IRQn" interrupt Be aware, disabling shared interrupt may affect other IPs

6.35.2.2 HAL_DAC_MspInit()

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

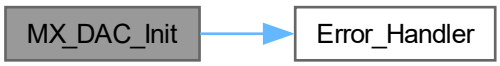
DAC GPIO Configuration PA4 ------> DAC_OUT1

6.35.2.3 MX_DAC_Init()

```
void MX_DAC_Init (
            void  )
```

DAC Initialization

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



Here is the caller graph for this function:

6.35.3 Variable Documentation

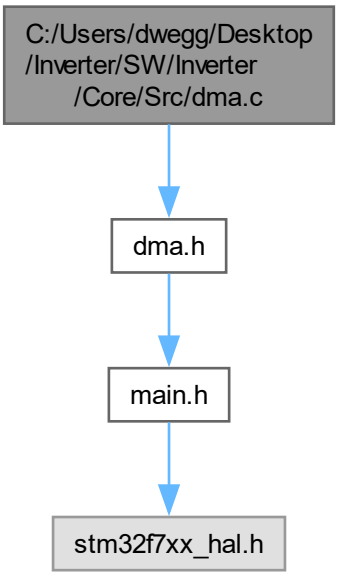
6.35.3.1 hdac

```
DAC_HandleTypeDef hdac
```

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

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

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



Functions

- void MX_DMA_Init (void)

6.36.1 Detailed Description

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

Attention

6.36.2 Function Documentation

6.36.2.1 MX_DMA_Init()

```
void MX_DMA_Init (
            void  )
```

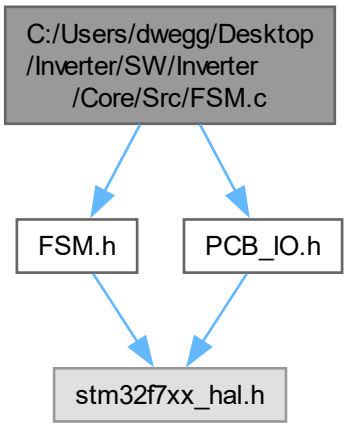
Enable DMA controller clock Here is the caller graph for this function:

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

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

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

Include dependency graph for FSM.c:



Functions

- void inv_init (InverterOperation *inv)

  Initialize the inverter operation structure.
- void inv_FSM (InverterOperation *inv)

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

6.37.1 Detailed Description

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

Attention

6.37.2 Function Documentation

6.37.2.1 inv_FSM()

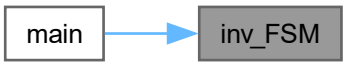
```
void inv_FSM (
            InverterOperation * inv )
```

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

Parameters

inv	Pointer to the inverter operation structure.

Here is the caller graph for this function:



6.37.2.2 inv_init()

```
void inv_init (
            InverterOperation * inv )
```

Initialize the inverter operation structure.

Parameters

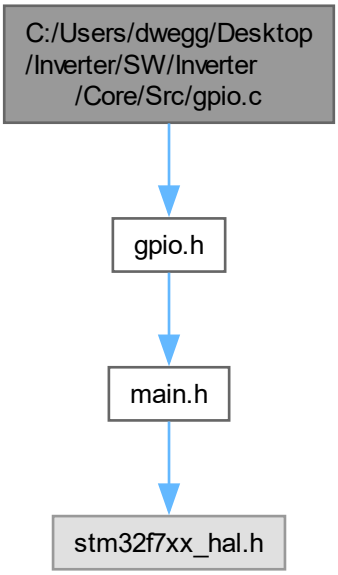
inv	Pointer to the inverter operation structure.

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

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

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

Include dependency graph for gpio.c:



Functions

- void MX_GPIO_Init (void)

6.38.1 Detailed Description

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

Attention

6.38.2 Function Documentation

6.38.2.1 MX_GPIO_Init()

```
void MX_GPIO_Init (
            void  )
```

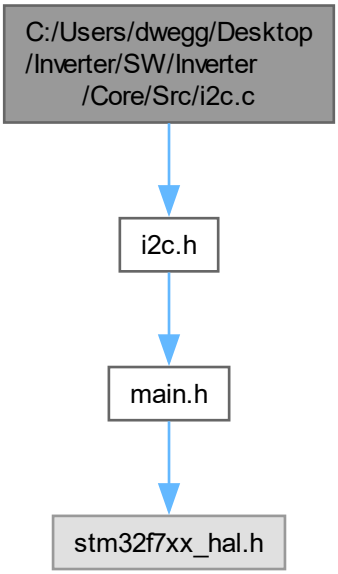
Configure pins as Analog Input Output EVENT_OUT EXTI PA9 ------> USB_OTG_FS_VBUS PA10 ------> USB↩ _OTG_FS_ID PA11 ------> USB_OTG_FS_DM PA12 ------> USB_OTG_FS_DP Here is the caller graph for this function:

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

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

```
#include "i2c.h"
```

Include dependency graph for i2c.c:



Functions

- void MX_I2C1_Init (void)
- void HAL_I2C_MspInit (I2C_HandleTypeDef *i2cHandle)
- void HAL_I2C_MspDeInit (I2C_HandleTypeDef *i2cHandle)

Variables

- I2C_HandleTypeDef hi2c1

6.39.1 Detailed Description

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

Attention

6.39.2 Function Documentation

6.39.2.1 HAL_I2C_MspDeInit()

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

I2C1 GPIO Configuration PB6 ------> I2C1_SCL PB7 ------> I2C1_SDA

6.39.2.2 HAL_I2C_MspInit()

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

Initializes the peripherals clock

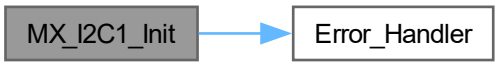
I2C1 GPIO Configuration PB6 ------> I2C1_SCL PB7 ------> I2C1_SDAHere is the call graph for this function:

6.39.2.3 MX_I2C1_Init()

```
void MX_I2C1_Init (
            void  )
```

Configure Analogue filter

Configure Digital filterHere is the call graph for this function:



Here is the caller graph for this function:

6.39.3 Variable Documentation

6.39.3.1 hi2c1

```
I2C_HandleTypeDef hi2c1
```

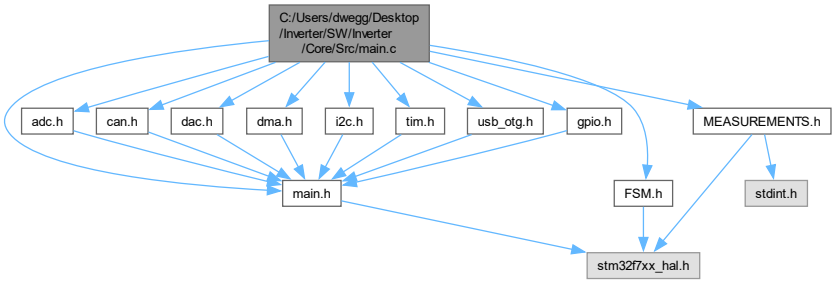
6.40 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/main.c File Reference

: 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 "MEASUREMENTS.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.

Variables

- ADC_HandleTypeDef hadc2
- ADC_HandleTypeDef hadc1
- TIM_HandleTypeDef htim1
- TIM_HandleTypeDef htim8
- InverterOperation invLeft
- InverterOperation invRight

6.40.1 Detailed Description

: Main program body

Attention

6.40.2 Function Documentation

6.40.2.1 Error_Handler()

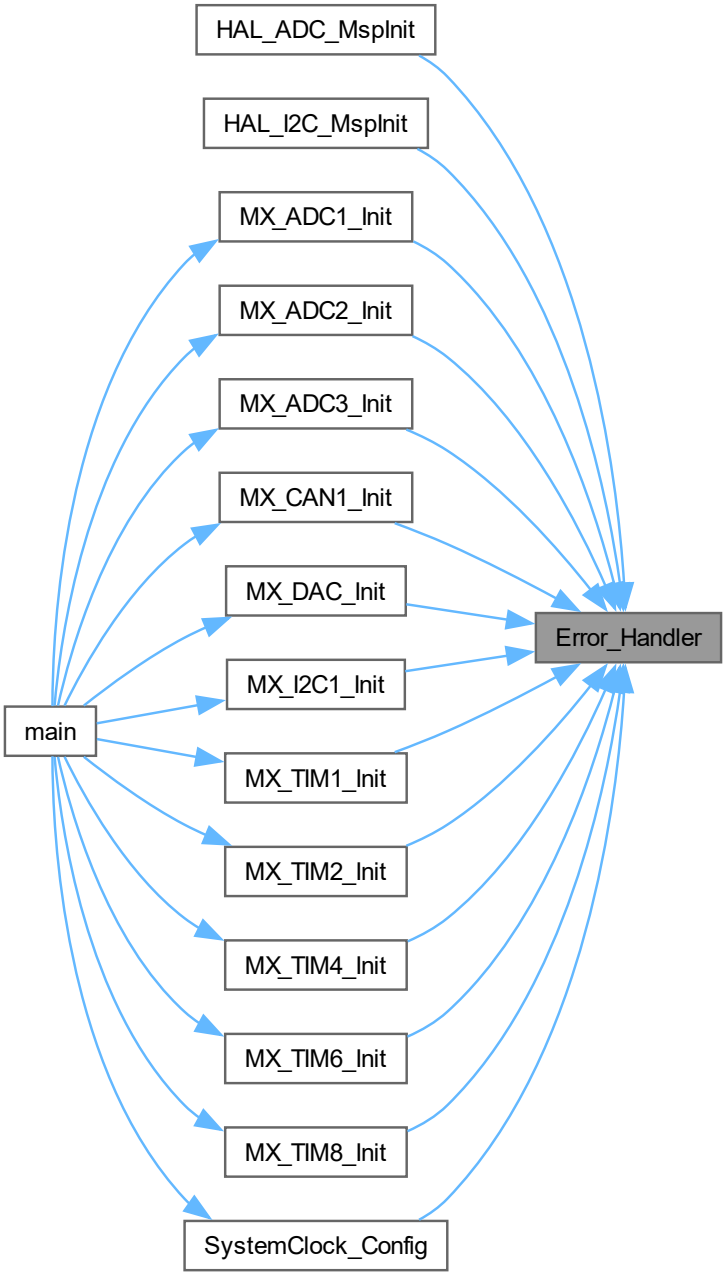
```
void Error_Handler (
            void  )
```

This function is executed in case of error occurrence.

Return values

None	

Here is the caller graph for this function:



6.40.2.2 main()

```
int main (
```

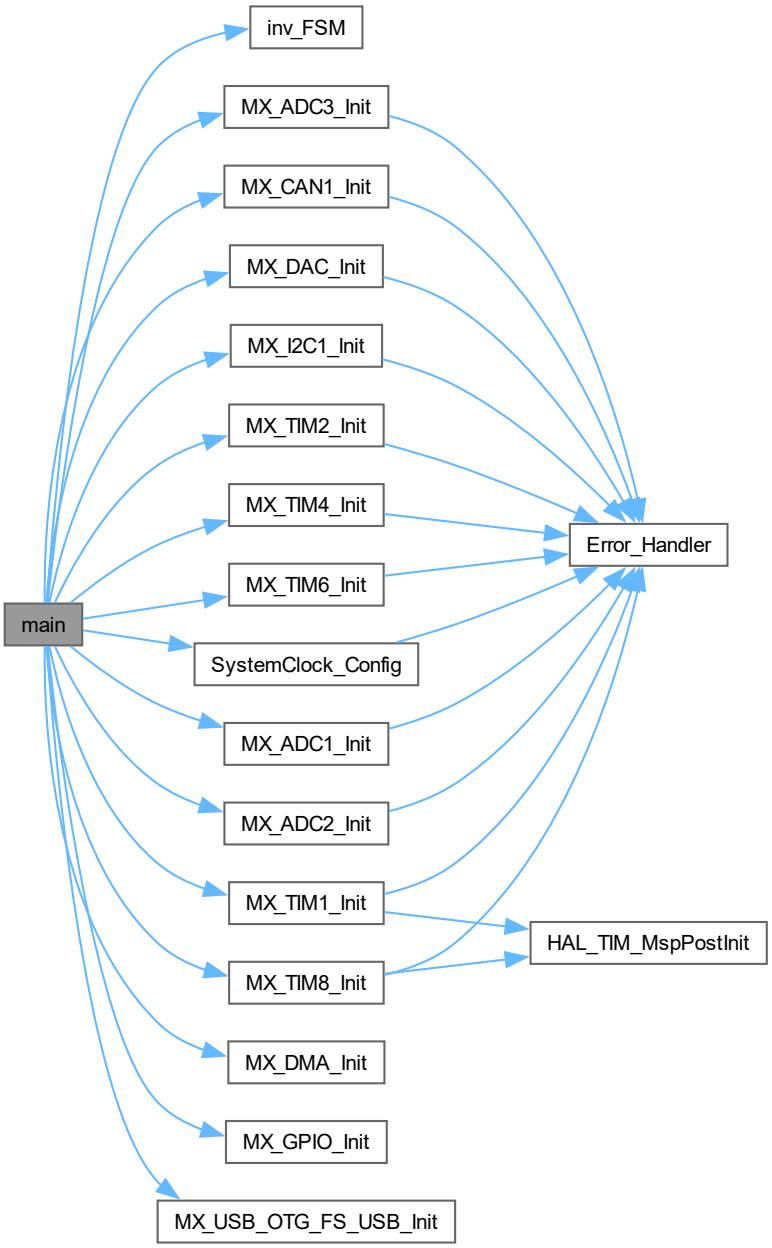
```
            void  )
```

The application entry point.

Return values

int	

Here is the call graph for this function:



6.40.2.3 SystemClock_Config()

```
void SystemClock_Config (
            void  )
```

System Clock Configuration.

Return values

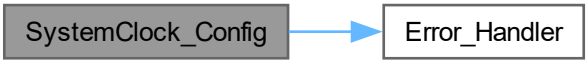
None	

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:

6.40.3 Variable Documentation

6.40.3.1 hadc1

```
ADC_HandleTypeDef hadc1  [extern]
```

6.40.3.2 hadc2

```
ADC_HandleTypeDef hadc2  [extern]
```

6.40.3.3 htim1

```
TIM_HandleTypeDef htim1  [extern]
```

6.40.3.4 htim8

```
TIM_HandleTypeDef htim8  [extern]
```

6.40.3.5 invLeft

```
InverterOperation invLeft
```

Initial value:

```
= {
    .state = INV_STATE_STARTUP,
    .LED_port = LED_LEFT_GPIO_Port,
    .LED_pin = LED_LEFT_Pin,
    .enable_port = ENABLE_L_GPIO_Port,
    .enable_pin = ENABLE_L_Pin
}
```

6.40.3.6 invRight

```
InverterOperation invRight
```

Initial value:

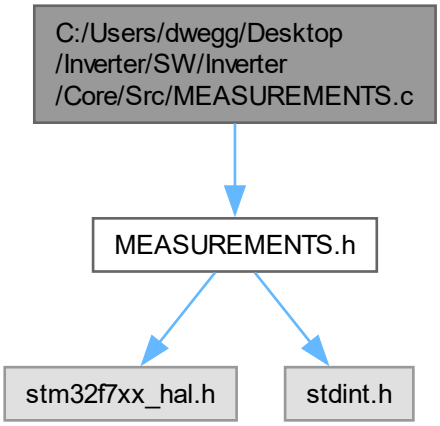
```
= {
    .state = INV_STATE_STARTUP,
    .LED_port = LED_RIGHT_GPIO_Port,
    .LED_pin = LED_RIGHT_Pin,
    .enable_port = ENABLE_R_GPIO_Port,
    .enable_pin = ENABLE_R_Pin
}
```

6.41 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/↩ MEASUREMENTS.c File Reference

This file provides functions for handling measurements.

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

Include dependency graph for MEASUREMENTS.c:



Functions

- uint8_t getADCelec (volatile uint32_t *ADC_raw, volatile Encoder *encoder, volatile Measurements *measurements)

  Get electrical ADC measurements.
- float getLinear (uint32_t bits, float slope, float offset)

  Convert ADC reading to physical measurement with linear response.

Variables

- volatile uint32_t ADC_LEFT_raw [4] = {0}

  Raw ADC readings for the left inverter.
- volatile uint32_t ADC_RIGHT_raw [4] = {0}

  Raw ADC readings for the right inverter.
- volatile Encoder encoder_LEFT = {0}

  Encoder data for the left inverter.
- volatile Encoder encoder_RIGHT = {0}

  Encoder data for the right inverter.
- volatile Measurements measurements_LEFT = {0}

  Measurements data for the left inverter.
- volatile Measurements measurements_RIGHT = {0}

  Measurements data for the right inverter.

6.41.1 Detailed Description

This file provides functions for handling measurements.

Attention

6.41.2 Function Documentation

6.41.2.1 getADCelec()

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

Get electrical ADC measurements.

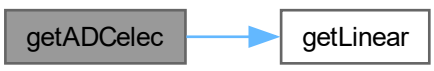
Parameters

ADC_raw	Pointer to the raw ADC values array.
encoder	Pointer to the encoder struct.
measurements	Pointer to the measurements struct to store the results.

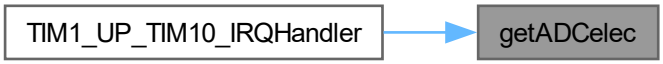
Return values

OK	0 if an error occurred, 1 if successful.

Here is the call graph for this function:



Here is the caller graph for this function:



6.41.2.2 getLinear()

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

Convert ADC reading to physical measurement with linear response.

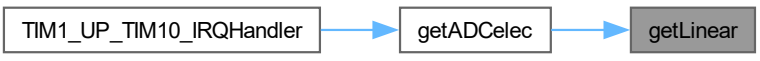
Parameters

bits	The ADC reading.
slope	The slope (units per volt).
offset	The offset (volts at zero).

Return values

measurement	The physical measurement.

Here is the caller graph for this function:



6.41.3 Variable Documentation

6.41.3.1 ADC_LEFT_raw

```
volatile uint32_t ADC_LEFT_raw[4] = {0}
```

Raw ADC readings for the left inverter.

6.41.3.2 ADC_RIGHT_raw

```
volatile uint32_t ADC_RIGHT_raw[4] = {0}
```

Raw ADC readings for the right inverter.

6.41.3.3 encoder_LEFT

```
volatile Encoder encoder_LEFT = {0}
```

Encoder data for the left inverter.

6.41.3.4 encoder_RIGHT

```
volatile Encoder encoder_RIGHT = {0}
```

Encoder data for the right inverter.

6.41.3.5 measurements_LEFT

```
volatile Measurements measurements_LEFT = {0}
```

Measurements data for the left inverter.

6.41.3.6 measurements_RIGHT

```
volatile Measurements measurements_RIGHT = {0}
```

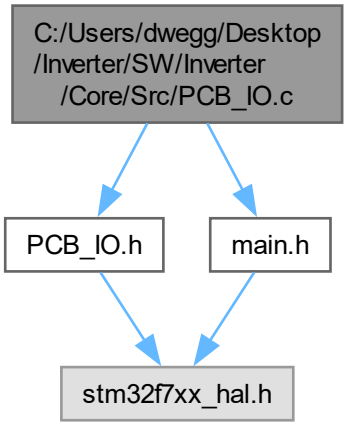
Measurements data for the right inverter.

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

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

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

Include dependency graph for PCB_IO.c:



Functions

- void LED_handler (LED *led, uint32_t ms_counter)

  LED handler function.

Variables

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

6.42.1 Detailed Description

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

Attention

6.42.2 Function Documentation

6.42.2.1 LED_handler()

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

LED handler function.

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

Parameters

led	Pointer to the LED structure.
ms_counter	Current millisecond counter.

Here is the caller graph for this function:

6.42.3 Variable Documentation

6.42.3.1 led_error

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

6.42.3.2 led_left

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

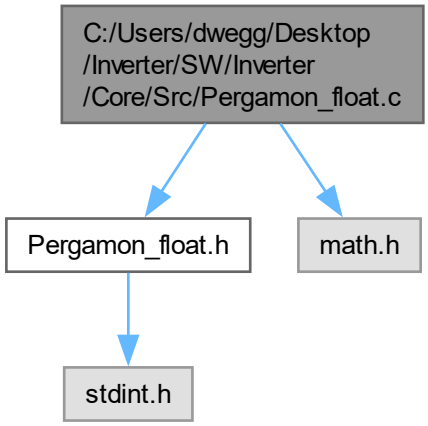
6.42.3.3 led_right

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

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

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

Include dependency graph for Pergamon_float.c:



Functions

- void pi_aw_calc (volatile pi_aw_struct *v)
- void pi_init (volatile pi_struct *v)
- void pi_calc (volatile pi_struct *v)
- void pi_extsat_calc (volatile pi_struct *v)
- void clarke3F_calc (volatile clarke3F_struct *v)
- void iclarke3F_calc (volatile iclarke3F_struct *v)
- void rot_calc (volatile rot_struct *v)
- void irot_calc (volatile irot_struct *v)
- void angle_calc (volatile angle_struct *v)
- void svpwm_calc (volatile svpwm_struct *v)
- void rampa_calc (volatile rampa_struct *v)
- void rampa_dual_calc (volatile rampa_dual_struct *v)
- void datalog_calc (volatile datalog_struct *dl)
- void filtreLP_init (volatile filtreLP_struct *v)
- void filtreLP_calc (volatile filtreLP_struct *v)
- void avg_calc_10_samples (volatile avg_struct_10 *v)
- void RMS_calc (volatile RMS_struct *v)
- void step_calc (volatile step_struct *v)

6.43.1 Function Documentation

6.43.1.1 angle_calc()

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

6.43.1.2 avg_calc_10_samples()

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

6.43.1.3 clarke3F_calc()

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

6.43.1.4 datalog_calc()

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

6.43.1.5 filtreLP_calc()

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

6.43.1.6 filtreLP_init()

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

6.43.1.7 iclarke3F_calc()

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

6.43.1.8 irot_calc()

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

6.43.1.9 pi_aw_calc()

```
void pi_aw_calc (
            volatile pi_aw_struct * v )
```

6.43.1.10 pi_calc()

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

6.43.1.11 pi_extsat_calc()

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

6.43.1.12 pi_init()

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

6.43.1.13 rampa_calc()

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

6.43.1.14 rampa_dual_calc()

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

6.43.1.15 RMS_calc()

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

6.43.1.16 rot_calc()

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

6.43.1.17 step_calc()

```
void step_calc (
            volatile step_struct * v )
```

6.43.1.18 svpwm_calc()

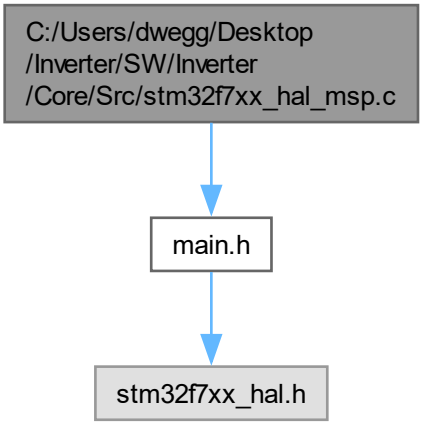
```
void svpwm_calc (
            volatile svpwm_struct * v )
```

6.44 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/stm32f7xx_hal_msp.c File Reference

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

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

Include dependency graph for stm32f7xx_hal_msp.c:



Functions

- void HAL_MspInit (void)

6.44.1 Detailed Description

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

Attention

6.44.2 Function Documentation

6.44.2.1 HAL_MspInit()

```
void HAL_MspInit (
            void  )
```

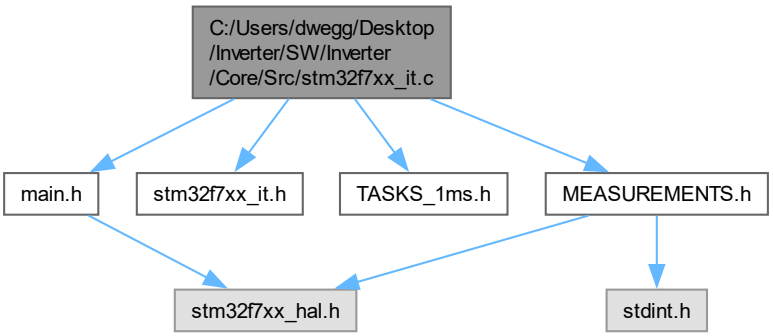
Initializes the Global MSP.

6.45 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/stm32f7xx_it.c File Reference

Interrupt Service Routines.

```
#include "main.h"
#include "stm32f7xx_it.h"
#include "TASKS_1ms.h"
#include "MEASUREMENTS.h"
```

Include 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 CAN1_RX1_IRQHandler (void)

  This function handles CAN1 RX1 interrupt.

- void TIM1_BRK_TIM9_IRQHandler (void)

  This function handles TIM1 break interrupt and TIM9 global interrupt.
- void TIM1_UP_TIM10_IRQHandler (void)

  This function handles TIM1 update interrupt and TIM10 global interrupt.
- void TIM1_TRG_COM_TIM11_IRQHandler (void)

  This function handles TIM1 trigger and commutation interrupts and TIM11 global interrupt.
- void TIM1_CC_IRQHandler (void)

  This function handles TIM1 capture compare 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

6.45.1 Detailed Description

Interrupt Service Routines.

Attention

6.45.2 Function Documentation

6.45.2.1 BusFault_Handler()

```
void BusFault_Handler (
             void  )
```

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

6.45.2.2 CAN1_RX0_IRQHandler()

```
void CAN1_RX0_IRQHandler (
            void  )
```

This function handles CAN1 RX0 interrupts.

6.45.2.3 CAN1_RX1_IRQHandler()

```
void CAN1_RX1_IRQHandler (
            void  )
```

This function handles CAN1 RX1 interrupt.

6.45.2.4 DebugMon_Handler()

```
void DebugMon_Handler (
            void  )
```

This function handles Debug monitor.

6.45.2.5 DMA2_Stream0_IRQHandler()

```
void DMA2_Stream0_IRQHandler (
            void  )
```

This function handles DMA2 stream0 global interrupt.

6.45.2.6 DMA2_Stream1_IRQHandler()

```
void DMA2_Stream1_IRQHandler (
            void  )
```

This function handles DMA2 stream1 global interrupt.

6.45.2.7 DMA2_Stream2_IRQHandler()

```
void DMA2_Stream2_IRQHandler (
            void  )
```

This function handles DMA2 stream2 global interrupt.

6.45.2.8 HardFault_Handler()

```
void HardFault_Handler (
            void  )
```

This function handles Hard fault interrupt.

6.45.2.9 MemManage_Handler()

```
void MemManage_Handler (
             void  )
```

This function handles Memory management fault.

6.45.2.10 NMI_Handler()

```
void NMI_Handler (
             void  )
```

This function handles Non maskable interrupt.

6.45.2.11 PendSV_Handler()

```
void PendSV_Handler (
             void  )
```

This function handles Pendable request for system service.

6.45.2.12 SVC_Handler()

```
void SVC_Handler (
             void  )
```

This function handles System service call via SWI instruction.

6.45.2.13 SysTick_Handler()

```
void SysTick_Handler (
             void  )
```

This function handles System tick timer.

6.45.2.14 TIM1_BRK_TIM9_IRQHandler()

```
void TIM1_BRK_TIM9_IRQHandler (
             void  )
```

This function handles TIM1 break interrupt and TIM9 global interrupt.

6.45.2.15 TIM1_CC_IRQHandler()

```
void TIM1_CC_IRQHandler (
             void  )
```

This function handles TIM1 capture compare interrupt.

6.45.2.16 TIM1_TRG_COM_TIM11_IRQHandler()

```
void TIM1_TRG_COM_TIM11_IRQHandler (
            void  )
```

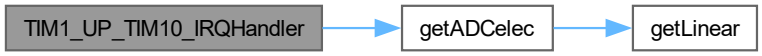
This function handles TIM1 trigger and commutation interrupts and TIM11 global interrupt.

6.45.2.17 TIM1_UP_TIM10_IRQHandler()

```
void TIM1_UP_TIM10_IRQHandler (
            void  )
```

This function handles TIM1 update interrupt and TIM10 global interrupt.

Here is the call graph for this function:

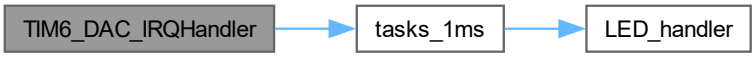


6.45.2.18 TIM6_DAC_IRQHandler()

```
void TIM6_DAC_IRQHandler (
            void  )
```

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

Here is the call graph for this function:



6.45.2.19 UsageFault_Handler()

```
void UsageFault_Handler (
            void  )
```

This function handles Undefined instruction or illegal state.

6.45.3 Variable Documentation

6.45.3.1 hcan1

```
CAN_HandleTypeDef hcan1  [extern]
```

6.45.3.2 hdac

```
DAC_HandleTypeDef hdac  [extern]
```

6.45.3.3 hdma_adc1

```
DMA_HandleTypeDef hdma_adc1  [extern]
```

6.45.3.4 hdma_adc2

```
DMA_HandleTypeDef hdma_adc2  [extern]
```

6.45.3.5 hdma_adc3

```
DMA_HandleTypeDef hdma_adc3  [extern]
```

6.45.3.6 htim1

```
TIM_HandleTypeDef htim1  [extern]
```

6.45.3.7 htim6

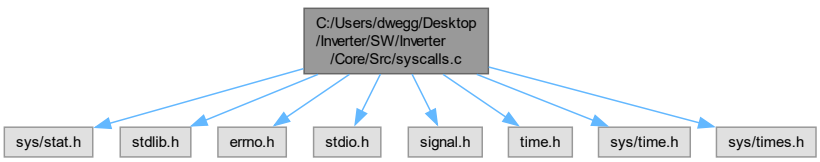
```
TIM_HandleTypeDef htim6  [extern]
```

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

STM32CubeIDE Minimal System calls file.

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

Include dependency graph for syscalls.c:



Functions

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

Variables

- char ** environ = __env

6.46.1 Detailed Description

STM32CubeIDE Minimal System calls file.

Author

Auto-generated by STM32CubeIDE

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

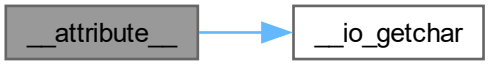
Attention

6.46.2 Function Documentation

6.46.2.1 __attribute__()

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

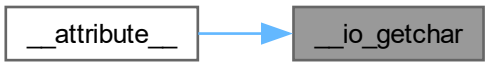
Here is the call graph for this function:



6.46.2.2 __io_getchar()

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

Here is the caller graph for this function:



6.46.2.3 __io_putchar()

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

6.46.2.4 _close()

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

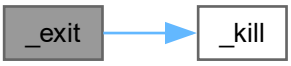
6.46.2.5 _execve()

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

6.46.2.6 _exit()

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

Here is the call graph for this function:



6.46.2.7 _fork()

```
int _fork (
              void  )
```

6.46.2.8 _fstat()

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

6.46.2.9 _getpid()

```
int _getpid (
              void  )
```

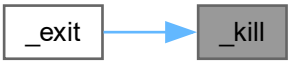
6.46.2.10 _isatty()

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

6.46.2.11 _kill()

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

Here is the caller graph for this function:



6.46.2.12 _link()

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

6.46.2.13 _lseek()

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

6.46.2.14 _open()

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

6.46.2.15 _stat()

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

6.46.2.16 _times()

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

6.46.2.17 _unlink()

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

6.46.2.18 _wait()

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

6.46.2.19 initialise_monitor_handles()

```
void initialise_monitor_handles ( )
```

6.46.3 Variable Documentation

6.46.3.1 environ

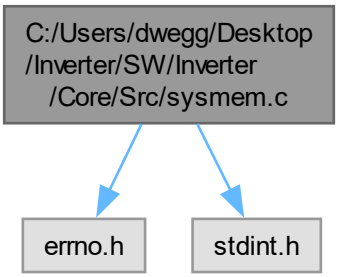
```
char** environ = __env
```

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

STM32CubeIDE System Memory calls file.

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

Include dependency graph for sysmem.c:



Functions

- void * _sbrk (ptrdiff_t incr)

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

6.47.1 Detailed Description

STM32CubeIDE System Memory calls file.

Author

Generated by STM32CubeIDE

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

Attention

6.47.2 Function Documentation

6.47.2.1 _sbrk()

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

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

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

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

Parameters

incr	Memory size

Returns

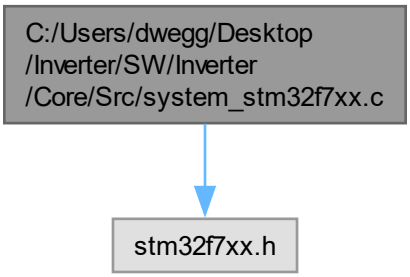
Pointer to allocated memory

6.48 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/system_stm32f7xx.c File Reference

CMSIS Cortex-M7 Device Peripheral Access Layer System Source File.

```
#include "stm32f7xx.h"
```

Include dependency graph for system_stm32f7xx.c:



Macros

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

Functions

- void SystemInit (void)

  Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.
- void SystemCoreClockUpdate (void)

  Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

Variables

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

6.48.1 Detailed Description

CMSIS Cortex-M7 Device Peripheral Access Layer System Source File.

Author

MCD Application Team

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

- SystemInit(): This function is called at startup just after reset and before branch to main program. This call is made inside the "startup_stm32f7xx.s" file.
- SystemCoreClock variable: Contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.
- SystemCoreClockUpdate(): Updates the variable SystemCoreClock and must be called whenever the core clock is changed during program execution.

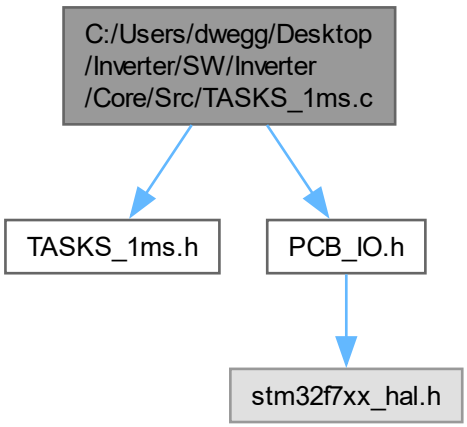
Attention

6.49 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/TASKS_1ms.c File Reference

This file contains functions to execute tasks every 1ms.

```
#include "TASKS_1ms.h"
#include "PCB_IO.h"
```

Include dependency graph for TASKS_1ms.c:



Functions

- void tasks_1ms (void)

  Function to be executed every 1ms.

6.49.1 Detailed Description

This file contains functions to execute tasks every 1ms.

Attention

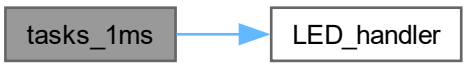
6.49.2 Function Documentation

6.49.2.1 tasks_1ms()

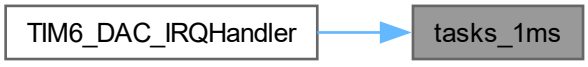
```
void tasks_1ms (
            void  )
```

Function to be executed every 1ms.

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



Here is the caller graph for this function:

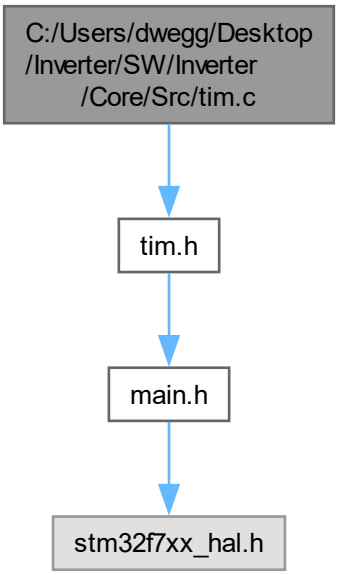


6.50 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/tim.c File Reference

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

```
#include "tim.h"
```

Include dependency graph for tim.c:



Functions

- void MX_TIM1_Init (void)
- void MX_TIM2_Init (void)
- void MX_TIM4_Init (void)
- void MX_TIM6_Init (void)
- void MX_TIM8_Init (void)
- void HAL_TIM_Base_MspInit (TIM_HandleTypeDef *tim_baseHandle)
- void HAL_TIM_IC_MspInit (TIM_HandleTypeDef *tim_icHandle)
- void HAL_TIM_MspPostInit (TIM_HandleTypeDef *timHandle)
- void HAL_TIM_Base_MspDeInit (TIM_HandleTypeDef *tim_baseHandle)
- void HAL_TIM_IC_MspDeInit (TIM_HandleTypeDef *tim_icHandle)

Variables

- TIM_HandleTypeDef htim1
- TIM_HandleTypeDef htim2
- TIM_HandleTypeDef htim4
- TIM_HandleTypeDef htim6
- TIM_HandleTypeDef htim8

6.50.1 Detailed Description

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

Attention

6.50.2 Function Documentation

6.50.2.1 HAL_TIM_Base_MspDeInit()

```
void HAL_TIM_Base_MspDeInit (
            TIM_HandleTypeDef * tim_baseHandle )
```

Uncomment the line below to disable the "TIM6_DAC_IRQn" interrupt Be aware, disabling shared interrupt may affect other IPs

6.50.2.2 HAL_TIM_Base_MspInit()

```
void HAL_TIM_Base_MspInit (
            TIM_HandleTypeDef * tim_baseHandle )
```

6.50.2.3 HAL_TIM_IC_MspDeInit()

```
void HAL_TIM_IC_MspDeInit (
            TIM_HandleTypeDef * tim_icHandle )
```

TIM2 GPIO Configuration PB10 ------> TIM2_CH3 PA15 ------> TIM2_CH1

TIM4 GPIO Configuration PD12 ------> TIM4_CH1 PD14 ------> TIM4_CH3

6.50.2.4 HAL_TIM_IC_MspInit()

```
void HAL_TIM_IC_MspInit (
            TIM_HandleTypeDef * tim_icHandle )
```

TIM2 GPIO Configuration PB10 ------> TIM2_CH3 PA15 ------> TIM2_CH1

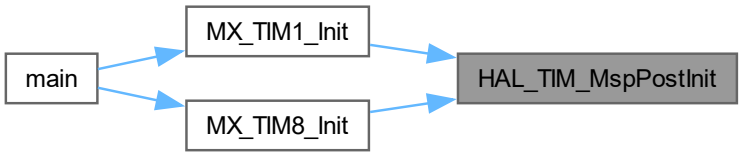
TIM4 GPIO Configuration PD12 ------> TIM4_CH1 PD14 ------> TIM4_CH3

6.50.2.5 HAL_TIM_MspPostInit()

```
void HAL_TIM_MspPostInit (
            TIM_HandleTypeDef * timHandle )
```

TIM1 GPIO Configuration PE8 ------> TIM1_CH1N PE9 ------> TIM1_CH1 PE10 ------> TIM1_CH2N PE11 ------> TIM1_CH2 PE12 ------> TIM1_CH3N PE13 ------> TIM1_CH3

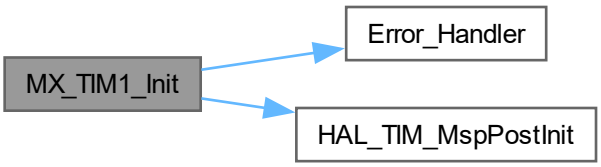
TIM8 GPIO Configuration PA5 ------> TIM8_CH1N PB14 ------> TIM8_CH2N PB15 ------> TIM8_CH3N PC6 ------> TIM8_CH1 PC7 ------> TIM8_CH2 PC8 ------> TIM8_CH3Here is the caller graph for this function:



6.50.2.6 MX_TIM1_Init()

```
void MX_TIM1_Init (
            void  )
```

Here is the call graph for this function:

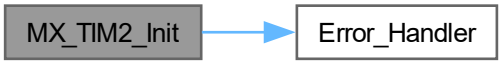


Here is the caller graph for this function:

6.50.2.7 MX_TIM2_Init()

```
void MX_TIM2_Init (
            void  )
```

Here is the call graph for this function:

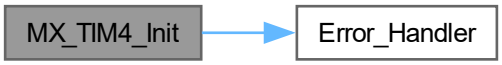


Here is the caller graph for this function:

6.50.2.8 MX_TIM4_Init()

```
void MX_TIM4_Init (
            void  )
```

Here is the call graph for this function:

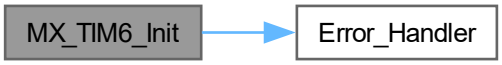


Here is the caller graph for this function:

6.50.2.9 MX_TIM6_Init()

```
void MX_TIM6_Init (
             void  )
```

Here is the call graph for this function:

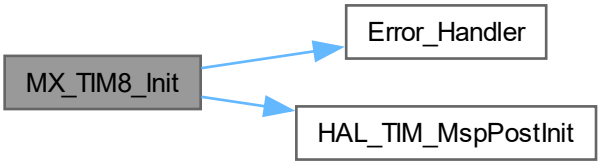


Here is the caller graph for this function:

6.50.2.10 MX_TIM8_Init()

```
void MX_TIM8_Init (
             void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:

6.50.3 Variable Documentation

6.50.3.1 htim1

```
TIM_HandleTypeDef htim1
```

6.50.3.2 htim2

```
TIM_HandleTypeDef htim2
```

6.50.3.3 htim4

```
TIM_HandleTypeDef htim4
```

6.50.3.4 htim6

```
TIM_HandleTypeDef htim6
```

6.50.3.5 htim8

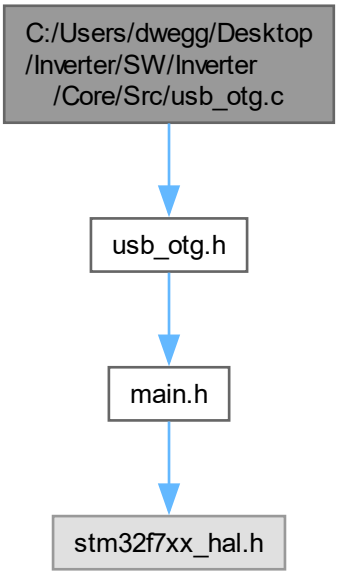
```
TIM_HandleTypeDef htim8
```

6.51 C:/Users/dwegg/Desktop/Inverter/SW/Inverter/Core/Src/usb_otg.c File Reference

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

```
#include "usb_otg.h"
```

Include dependency graph for usb_otg.c:



Functions

- void MX_USB_OTG_FS_USB_Init (void)

6.51.1 Detailed Description

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

Attention

6.51.2 Function Documentation

6.51.2.1 MX_USB_OTG_FS_USB_Init()

```
void MX_USB_OTG_FS_USB_Init (
            void  )
```

Here is the caller graph for this function:

Index