Firmware documentation - IMU board

Generated by Doxygen 1.8.13

Contents

Chapter 1

Firmware

This is the firmware of the IMU board.

Version

1.0

This is the firmware of the IMU board. It can read up to 17 IMU modules connected to the PSoC microcontroller.

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 documented files with brief descriptions:

Chapter 4

Data Structure Documentation

4.1 st_data Struct Reference

Data Fields

- uint8 **buffer** [128]
- int16 **length**
- int16 **ind**
- uint8 **ready**

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

- **globals.h**

4.2 st_imu Struct Reference

Data Fields

- uint8 **flags**
- int16 **accel_value** [3]
- int16 **gyro_value** [3]
- int16 **mag_value** [3]
- int16 **temp_value**

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

- **globals.h**

4.3 st_mem Struct Reference

Data Fields

- uint8 **flag**
- uint8 **id**
- uint8 **baud_rate**
- uint8 **watchdog_period**

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

- **globals.h**

Chapter 5

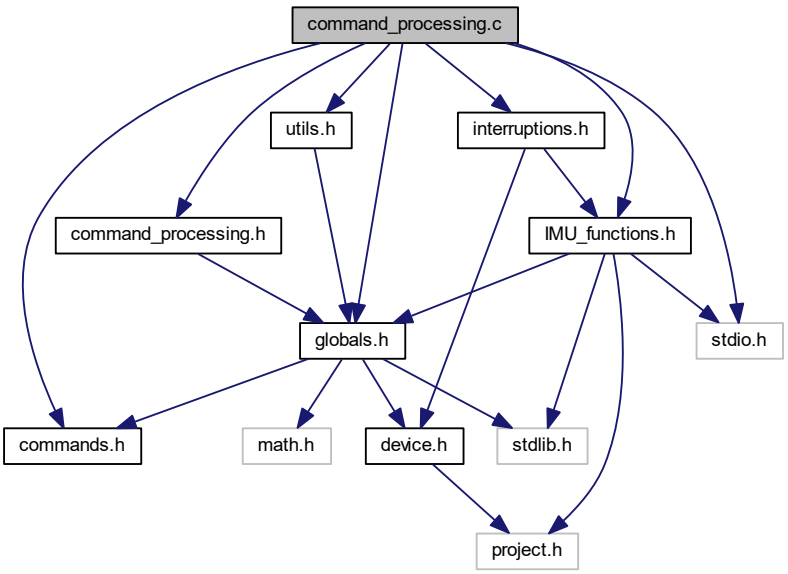
File Documentation

5.1 command_processing.c File Reference

Command processing functions.

```
#include <command_processing.h>
#include <interruptions.h>
#include <stdio.h>
#include <utils.h>
#include <IMU_functions.h>
#include <globals.h>
#include "commands.h"
```

Include dependency graph for command_processing.c:



Functions

- void **commProcess** ()
- void **infoSend** ()
- void **infoGet** (uint16 info_type)
- void **setZeros** ()
- void **get_param_list** (uint16 index)
- void **infoPrepare** (unsigned char ∗info_string)
- void **commWrite** (uint8 ∗packet_data, const uint16 packet_lenght)
- uint8 **LCRChecksum** (uint8 ∗data_array, uint8 data_length)
- void **sendAcknowledgment** (const uint8 value)
- uint8 **memStore** (int displacement)
- void **memRecall** ()
- uint8 **memRestore** ()
- uint8 **memInit** ()
- void **cmd_get_measurements** ()
- void **cmd_set_inputs** ()
- void **cmd_activate** ()
- void **cmd_get_activate** ()
- void **cmd_get_curr_and_meas** ()
- void **cmd_get_currents** ()
- void **cmd_set_baudrate** ()
- void **cmd_ping** ()
- void **cmd_set_watchdog** ()
- void **cmd_get_inputs** ()
- void **cmd_store_params** ()
- void **cmd_get_emg** ()
- void **cmd_get_imu_readings** ()

Variables

- reg8 ∗ **EEPROM_ADDR** = (reg8 ∗) CYDEV_EE_BASE

5.1.1 Detailed Description

Command processing functions.

Date

October 01, 2017

Author

Centro "E.Piaggio"

Copyright

5.1.2 Function Documentation

5.1.2.1 cmd_get_measurements()

```
void cmd_get_measurements ( )
```

Bunch of functions used on request from UART communication

5.1.2.2 memInit()

```
uint8 memInit ( )
```

This function initialize memory when eeprom is compromised.

5.1.2.3 memRecall()

```
void memRecall ( )
```

This function loads user settings from the eeprom.

5.1.2.4 memRestore()

```
uint8 memRestore ( )
```

This function loads default settings from the eeprom.

5.1.2.5 memStore()

```
uint8 memStore (
            int displacement )
```

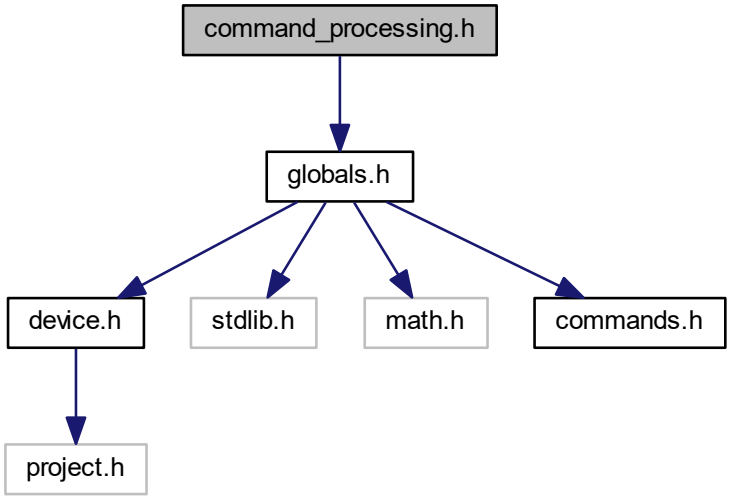
This function stores current memory settings on the eeprom with the specified displacement

5.2 command_processing.h File Reference

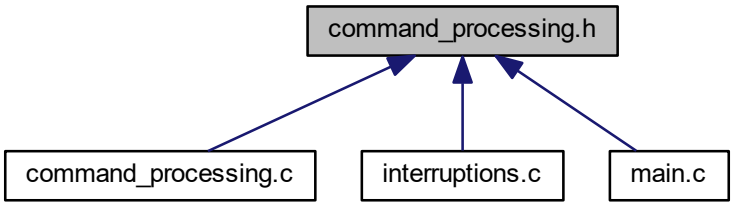
Definition of command processing functions.

```
#include <globals.h>
```

Include dependency graph for command_processing.h:



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



Functions

- void **setZeros** (void)
- void **get_param_list** (uint16 index)
- void **infoPrepare** (unsigned char ∗)
- void **infoSend** ()
- void **infoGet** (uint16)

- void **commProcess** ()
- void **commWrite** (uint8 ∗, const uint16)
- uint8 **memStore** (int)
- void **sendAcknowledgment** (const uint8)
- void **memRecall** ()
- uint8 **memRestore** ()
- uint8 **memInit** ()
- uint8 **LCRChecksum** (uint8 ∗, uint8)
- void **cmd_activate** ()
- void **cmd_set_inputs** ()
- void **cmd_get_measurements** ()
- void **cmd_get_currents** ()
- void **cmd_get_emg** ()
- void **cmd_set_watchdog** ()
- void **cmd_get_activate** ()
- void **cmd_set_baudrate** ()
- void **cmd_get_inputs** ()
- void **cmd_store_params** ()
- void **cmd_ping** ()
- void **cmd_get_imu_readings** ()

5.2.1 Detailed Description

Definition of command processing functions.

Date

October 01, 2017

Author

Centro "E.Piaggio"

Copyright

5.2.2 Function Documentation

5.2.2.1 cmd_get_measurements()

```
void cmd_get_measurements ( )
```

Bunch of functions used on request from UART communication

5.2.2.2 memInit()

```
uint8 memInit ( )
```

This function initialize memory when eeprom is compromised.

5.2.2.3 memRecall()

```
void memRecall ( )
```

This function loads user settings from the eeprom.

5.2.2.4 memRestore()

```
uint8 memRestore ( )
```

This function loads default settings from the eeprom.

5.2.2.5 memStore()

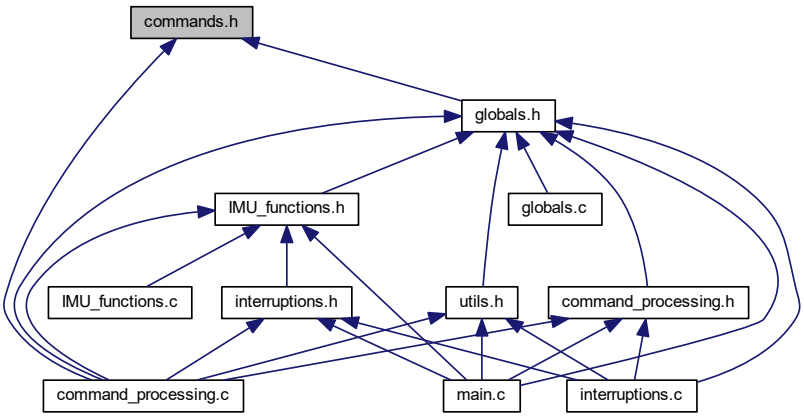
```
uint8 memStore (
            int displacement )
```

This function stores current memory settings on the eeprom with the specified displacement

5.3 commands.h File Reference

Definitions for commands, parameters and packages.

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



Macros

QB Move Information Strings

- #define **INFO_ALL** 0
  All system information.

Enumerations

QB Move Commands

- enum **qbmove_command** {
  CMD_PING = 0, **CMD_SET_ZEROS** = 1, **CMD_STORE_PARAMS** = 3, **CMD_STORE_DEFAULT_P↩ ARAMS** = 4,
  CMD_RESTORE_PARAMS = 5, **CMD_GET_INFO** = 6, **CMD_SET_VALUE** = 7, **CMD_GET_VALUE** = 8,
  CMD_BOOTLOADER = 9, **CMD_INIT_MEM** = 10, **CMD_CALIBRATE** = 11, **CMD_GET_PARAM_LIST** = 12,
  CMD_HAND_CALIBRATE = 13, **CMD_ACTIVATE** = 128, **CMD_GET_ACTIVATE** = 129, **CMD_SET↩ _INPUTS** = 130,
  CMD_GET_INPUTS = 131, **CMD_GET_MEASUREMENTS** = 132, **CMD_GET_CURRENTS** = 133, **C↩ MD_GET_CURR_AND_MEAS** = 134,
  CMD_SET_POS_STIFF = 135, **CMD_GET_EMG** = 136, **CMD_SET_WATCHDOG** = 138, **CMD_SET↩ _BAUDRATE** = 139,
  CMD_GET_N_IMU = 160, **CMD_GET_IMU_READINGS** = 161 }

QB Move Parameters

- #define **PARAM_BYTE_SLOT** 50
- #define **PARAM_MENU_SLOT** 150
- enum **qbmove_parameter** { **PARAM_ID** = 0 }
- enum **qbmove_resolution** {
  RESOLUTION_360 = 0, **RESOLUTION_720** = 1, **RESOLUTION_1440** = 2, **RESOLUTION_2880** = 3,
  RESOLUTION_5760 = 4, **RESOLUTION_11520** = 5, **RESOLUTION_23040** = 6, **RESOLUTION_46080** = 7,
  RESOLUTION_92160 = 8 }
- enum **qbmove_input_mode** {
  INPUT_MODE_EXTERNAL = 0, **INPUT_MODE_ENCODER3** = 1, **INPUT_MODE_EMG_PROPORTION↩ AL** = 2, **INPUT_MODE_EMG_INTEGRAL** = 3,
  INPUT_MODE_EMG_FCFS = 4, **INPUT_MODE_EMG_FCFS_ADV** = 5 }
- enum **qbmove_control_mode** { **CONTROL_ANGLE** = 0, **CONTROL_PWM** = 1, **CONTROL_CURRENT** = 2, **CURR_AND_POS_CONTROL** = 3 }
- enum **motor_supply_tipe** { **MAXON_24V** = 0, **MAXON_12V** = 1 }
- enum **acknowledgment_values** { **ACK_ERROR** = 0, **ACK_OK** = 1 }
- enum **data_types** {
  TYPE_FLAG = 0, **TYPE_INT8** = 1, **TYPE_UINT8** = 2, **TYPE_INT16** = 3,
  TYPE_UINT16 = 4, **TYPE_INT32** = 5, **TYPE_UINT32** = 6, **TYPE_FLOAT** = 7,
  TYPE_DOUBLE = 8 }

5.3.1 Detailed Description

Definitions for commands, parameters and packages.

Date

October 01, 2017

Author

Centro "E.Piaggio"

Copyright

5.3.2 Enumeration Type Documentation

5.3.2.1 qbmove_command

enum **qbmove_command**

Enumerator

CMD_PING	Asks for a ping message.
CMD_SET_ZEROS	Command for setting the encoders zero position.
CMD_STORE_PARAMS	Stores all parameters in memory and loads them
CMD_STORE_DEFAULT_PARAMS	Store current parameters as factory parameters.
CMD_RESTORE_PARAMS	Restore default factory parameters.
CMD_GET_INFO	Asks for a string of information about.
CMD_SET_VALUE	Not Used.
CMD_GET_VALUE	Not Used.
CMD_BOOTLOADER	Sets the bootloader modality to update the firmware
CMD_INIT_MEM	Initialize the memory with the defalut values.
CMD_CALIBRATE	Starts the stiffness calibration of the qbMove or the hand closure and opening calibration
CMD_GET_PARAM_LIST	Command to get the parameters list or to set a defined value chosen by the use
CMD_HAND_CALIBRATE	Starts a series of opening and closures of the hand.
CMD_ACTIVATE	Command for activating/deactivating the device
CMD_GET_ACTIVATE	Command for getting device activation state
CMD_SET_INPUTS	Command for setting reference inputs.
CMD_GET_INPUTS	Command for getting reference inputs.
CMD_GET_MEASUREMENTS	Command for asking device's position measurements
CMD_GET_CURRENTS	Command for asking device's current measurements
CMD_GET_CURR_AND_MEAS	Command for asking device's measurements and currents
CMD_SET_WATCHDOG	Command for setting watchdog timer or disable it
CMD_SET_BAUDRATE	Command for setting baudrate
CMD_GET_N_IMU	of communication

5.3.2.2 qbmove_control_mode

enum **qbmove_control_mode**

Enumerator

CONTROL_ANGLE	Classic position control.
CONTROL_PWM	Direct PWM value.
CONTROL_CURRENT	Current control (beta)
CURR_AND_POS_CONTROL	Current control (beta)

5.3.2.3 qbmove_input_mode

enum **qbmove_input_mode**

Enumerator

INPUT_MODE_EXTERNAL	References through external commands (default)
INPUT_MODE_ENCODER3	Encoder 3 drives all inputs.
INPUT_MODE_EMG_PROPORTIONAL	Use EMG measure to proportionally drive the position of the motor 1
INPUT_MODE_EMG_INTEGRAL	Use 2 EMG signals to drive motor position
INPUT_MODE_EMG_FCFS	Use 2 EMG. First reaching threshold wins and its value defines hand closure
INPUT_MODE_EMG_FCFS_ADV	Use 2 EMG. First reaching threshold wins and its value defines hand closure Wait for both EMG to lower under threshold

5.3.2.4 qbmove_parameter

enum **qbmove_parameter**

Enumerator

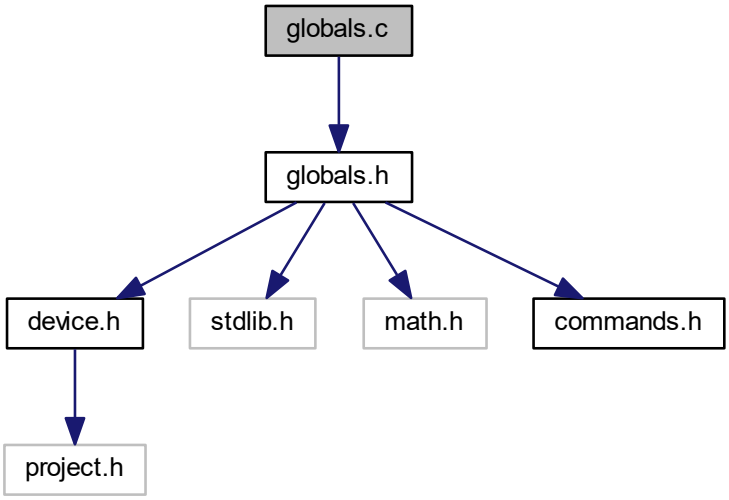
PARAM_ID	Device's ID number.

5.4 globals.c File Reference

Global variables.

```
#include <globals.h>
```

Include dependency graph for globals.c:



Variables

- struct **st_data g_rx**
- struct **st_mem** g_mem **c_mem**
- float **tau_feedback**
- uint32 **timer_value**
- uint32 **timer_value0**
- int32 **dev_tension**
- uint8 **dev_pwm_limit**
- CYBIT **reset_last_value_flag**
- CYBIT **tension_valid**
- CYBIT **interrupt_flag**
- CYBIT **watchdog_flag**
- int16 **ADC_buf** [1]
- int8 **pwm_sign**
- uint8 **N_IMU_Connected**
- uint8 **IMU_connected** [N_IMU_MAX]
- uint8 **IMU_conf** [N_IMU_MAX][5]
- int **imus_data_size**
- int **single_imu_size** [N_IMU_MAX]
- struct **st_imu g_imu** [N_IMU_MAX]
- struct **st_imu g_imuNew** [N_IMU_MAX]
- uint8 **Accel** [N_IMU_MAX][6]
- uint8 **Gyro** [N_IMU_MAX][6]
- uint8 **Mag** [N_IMU_MAX][6]
- uint8 **MagCal** [N_IMU_MAX][3]
- uint8 **Temp** [N_IMU_MAX][2]
- int **frsAccReg**
- int **frsGyroReg**
- uint8 **rateAcc**
- uint8 **rateGyro**

5.4.1 Detailed Description

Global variables.

Date

October 01, 2017

Author

Centro "E.Piaggio"

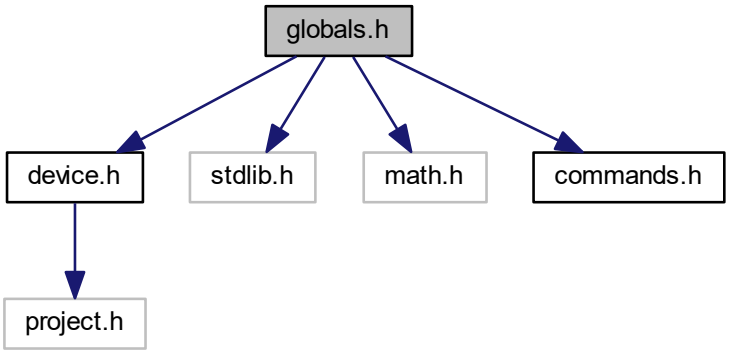
Copyright

5.5 globals.h File Reference

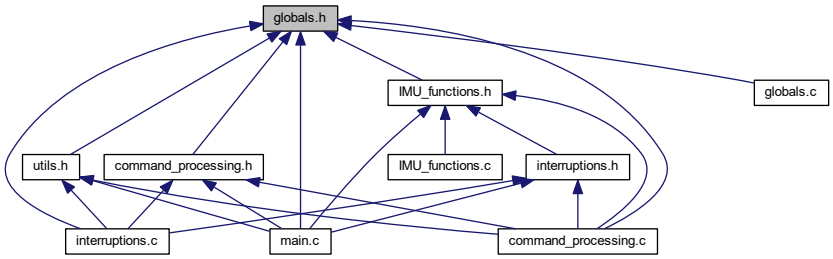
Global definitions and macros are set in this file.

```
#include <device.h>
#include "stdlib.h"
#include "math.h"
#include "commands.h"
```

Include dependency graph for globals.h:



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



Data Structures

- struct **st_data**
- struct **st_mem**
- struct **st_imu**

Macros

- #define **VERSION** "IMUboard v 1.0.1"
- #define **NUM_OF_ANALOG_INPUTS** 1
- #define **N_IMU_MAX** 17
- #define **CALIBRATION_DIV** 10
- #define **DIV_INIT_VALUE** 1
- #define **DMA_BYTES_PER_BURST** 2
- #define **DMA_REQUEST_PER_BURST** 1
- #define **DMA_SRC_BASE** (CYDEV_PERIPH_BASE)
- #define **DMA_DST_BASE** (CYDEV_SRAM_BASE)
- #define **WAIT_START** 0
- #define **WAIT_ID** 1
- #define **WAIT_LENGTH** 2
- #define **RECEIVE** 3
- #define **UNLOAD** 4
- #define **FALSE** 0
- #define **TRUE** 1
- #define **DEFAULT_EEPROM_DISPLACEMENT** 8
- #define **MAX_WATCHDOG_TIMER** 250

Variables

- struct **st_data g_rx**
- struct **st_mem** g_mem **c_mem**
- uint32 **timer_value**
- uint32 **timer_value0**
- int32 **dev_tension**
- uint8 **dev_pwm_limit**
- CYBIT **reset_last_value_flag**
- CYBIT **tension_valid**

- CYBIT **interrupt_flag**
- CYBIT **watchdog_flag**
- float **tau_feedback**
- int16 **ADC_buf** [1]
- int8 **pwm_sign**
- uint8 **N_IMU_Connected**
- uint8 **IMU_connected** [N_IMU_MAX]
- uint8 **IMU_conf** [N_IMU_MAX][5]
- int **imus_data_size**
- int **single_imu_size** [N_IMU_MAX]
- struct **st_imu g_imu** [N_IMU_MAX]
- struct **st_imu g_imuNew** [N_IMU_MAX]
- uint8 **Accel** [N_IMU_MAX][6]
- uint8 **Gyro** [N_IMU_MAX][6]
- uint8 **Mag** [N_IMU_MAX][6]
- uint8 **MagCal** [N_IMU_MAX][3]
- uint8 **Temp** [N_IMU_MAX][2]

5.5.1 Detailed Description

Global definitions and macros are set in this file.

Date

October 01, 2017

Author

Centro "E.Piaggio"

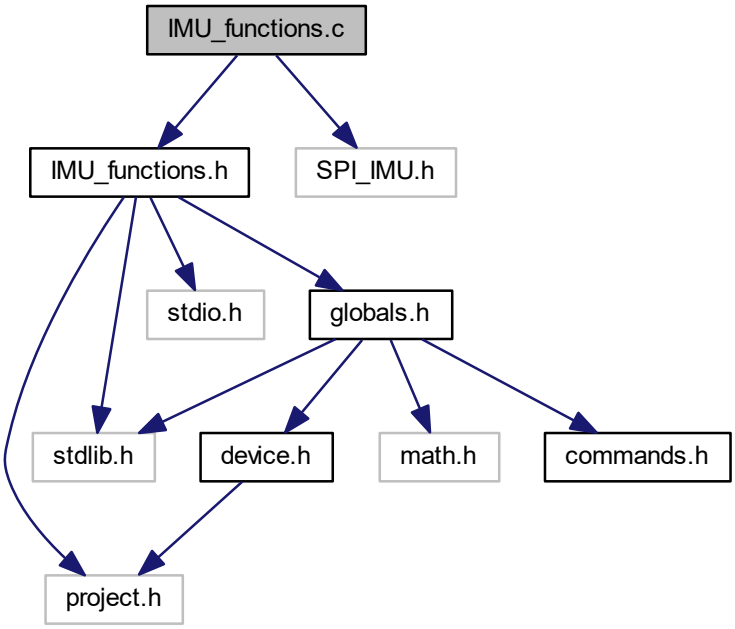
Copyright

5.6 IMU_functions.c File Reference

Implementation of IMU module functions.

```
#include <IMU_functions.h>
#include <SPI_IMU.h>
```

Include dependency graph for IMU_functions.c:



Functions

- void **ImusReset** ()
- void **InitIMU** ()
- void **InitIMUMagCal** ()
- void **ChipSelector** (int n)
- void **InitIMUgeneral** ()
- void **ReadIMU** (int n)
- void **ReadAcc** (int n)
- void **ReadGyro** (int n)
- void **ReadMag** (int n)
- void **ReadMagCal** (int n)
- void **ReadAllIMUs** ()
- void **ReadTemp** (int n)
- void **WriteControlRegister** (uint8 address, uint8 dta)

- uint8 **ReadControlRegister** (uint8 address)

Variables

- uint8 **Accel** [N_IMU_MAX][6]
- uint8 **Gyro** [N_IMU_MAX][6]
- uint8 **Mag** [N_IMU_MAX][6]
- uint8 **MagCal** [N_IMU_MAX][3]

5.6.1 Detailed Description

Implementation of IMU module functions.

Date

October 01, 2017

Author

Centro "E.Piaggio"

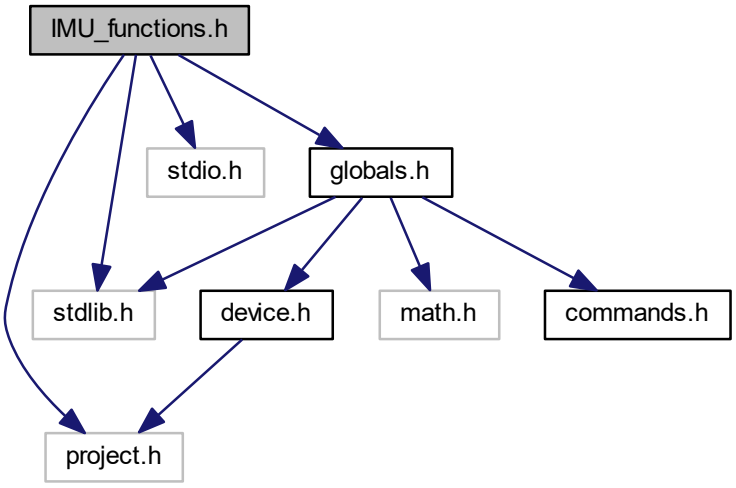
Copyright

5.7 IMU_functions.h File Reference

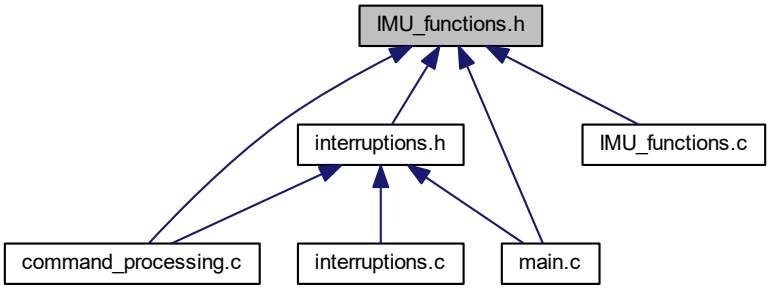
Definition of IMU module functions.

```
#include <project.h>
#include <stdlib.h>
#include <stdio.h>
#include <globals.h>
```

Include dependency graph for IMU_functions.h:



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



Macros

- #define **MPU9250_RCR** 0x80
- #define **MPU9250_WCR** 0x00
- #define **MPU9250_CONFIG** 0x1A
- #define **MPU9250_GYRO_CONFIG** 0x1B
- #define **MPU9250_ACCEL_CONFIG** 0x1C
- #define **MPU9250_ACCEL_CONFIG2** 0x1D
- #define **MPU9250_ACCEL_XOUT_H** 0x3B
- #define **MPU9250_ACCEL_XOUT_L** 0x3C
- #define **MPU9250_ACCEL_YOUT_H** 0x3D
- #define **MPU9250_ACCEL_YOUT_L** 0x3E
- #define **MPU9250_ACCEL_ZOUT_H** 0x3F
- #define **MPU9250_ACCEL_ZOUT_L** 0x40
- #define **MPU9250_TEMP_OUT_H** 0x41
- #define **MPU9250_TEMP_OUT_L** 0x42
- #define **MPU9250_GYRO_XOUT_H** 0x43
- #define **MPU9250_GYRO_XOUT_L** 0x44
- #define **MPU9250_GYRO_YOUT_H** 0x45
- #define **MPU9250_GYRO_YOUT_L** 0x46
- #define **MPU9250_GYRO_ZOUT_H** 0x47
- #define **MPU9250_GYRO_ZOUT_L** 0x48
- #define **MPU9250_USER_CTRL** 0x6A
- #define **MPU9250_PWR_MGMT_1** 0x6B
- #define **MPU9250_WHO_AM_I** 0x75
- #define **MPU9250_FIFO_EN** 0x23
- #define **MPU9250_I2C_MST_CTRL** 0x24
- #define **MPU9250_I2C_SLV0_ADDR** 0x25
- #define **MPU9250_I2C_SLV0_REG** 0x26
- #define **MPU9250_I2C_SLV0_CTRL** 0x27
- #define **MPU9250_I2C_SLV1_ADDR** 0x28
- #define **MPU9250_I2C_SLV1_REG** 0x29
- #define **MPU9250_I2C_SLV1_CTRL** 0x2A
- #define **MPU9250_EXT_SENS_DATA_00** 0x49
- #define **MPU9250_EXT_SENS_DATA_01** 0x4A
- #define **MPU9250_EXT_SENS_DATA_02** 0x4B

- #define **MPU9250_EXT_SENS_DATA_03** 0x4C
- #define **MPU9250_EXT_SENS_DATA_04** 0x4D
- #define **MPU9250_EXT_SENS_DATA_05** 0x4E
- #define **MPU9250_EXT_SENS_DATA_06** 0x4F
- #define **MPU9250_EXT_SENS_DATA_07** 0x50
- #define **MPU9250_I2C_SLV0_D0** 0x63
- #define **MPU9250_I2C_SLV1_D0** 0x64
- #define **MPU9250_I2C_MST_DELAY_CTRL** 0x67
- #define **AK8936_ADDRESS** 0x0C
- #define **AK8936_WIA** 0x00
- #define **AK8936_INFO** 0x01
- #define **AK8936_ST1** 0x02
- #define **AK8936_XOUT_L** 0x03
- #define **AK8936_XOUT_H** 0x04
- #define **AK8936_YOUT_L** 0x05
- #define **AK8936_YOUT_H** 0x06
- #define **AK8936_ZOUT_L** 0x07
- #define **AK8936_ZOUT_H** 0x08
- #define **AK8936_ST2** 0x09
- #define **AK8936_CNTL** 0x0A
- #define **AK8963_CNTL2** 0x0B
- #define **AK8936_ASTC** 0x0C
- #define **AK8936_I2CDIS** 0x0F
- #define **ACC_SF_2G** 0x00
- #define **ACC_SF_4G** 0x08
- #define **ACC_SF_8G** 0x10
- #define **ACC_SF_16G** 0x18
- #define **GYRO_SF_250** 0x00
- #define **GYRO_SF_500** 0x80
- #define **GYRO_SF_2000** 0x18
- #define **G_TO_MS2** 9.79
- #define **DEG_TO_RAD** (3.14159265359 / 180.0)

Functions

- void **ImusReset** ()
- void **InitIMU** ()
- void **InitIMUMagCal** ()
- void **InitIMUgeneral** ()
- void **ReadAcc** (int n)
- void **ReadGyro** (int n)
- void **ReadMag** (int n)
- void **ReadMagCal** (int n)
- void **ReadTemp** (int n)
- void **ReadIMU** (int n)
- void **ReadAllIMUs** ()
- void **LF_Frequency_Change_Accel_And_Gyro** (int d_frequency, int n_imu)
- void **LF_Frequency_Change_Accel** (int d_frequency, int n_imu)
- void **LF_Frequency_Change_Gyro** (int d_frequency, int n_imu)
- uint8 **ReadControlRegister** (uint8 address)
- void **WriteControlRegister** (uint8 address, uint8 dta)

- void **ChipSelector** (int n)

5.7.1 Detailed Description

Definition of IMU module functions.

Date

October 01, 2017

Author

Centro "E.Piaggio"

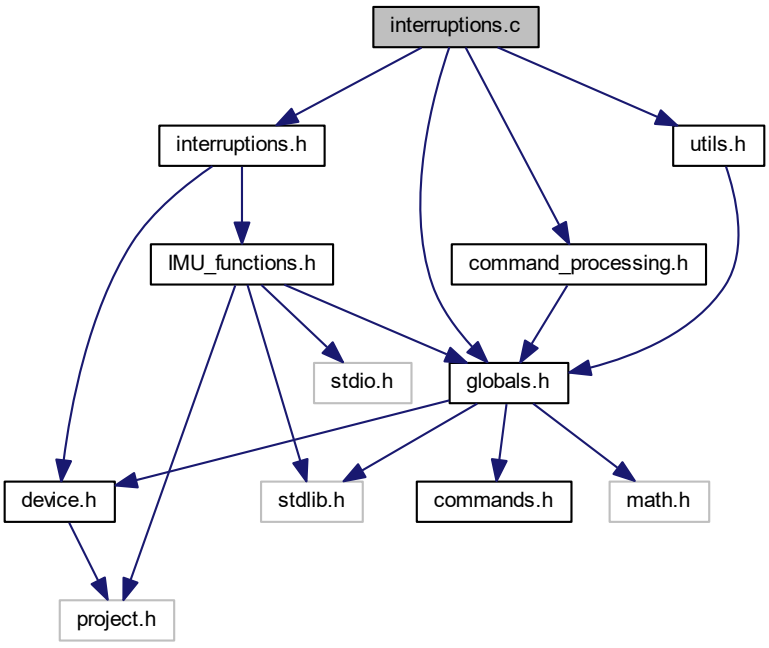
Copyright

5.8 interruptions.c File Reference

Interruption functions are in this file.

```
#include <interruptions.h>
#include <command_processing.h>
#include <globals.h>
#include <utils.h>
```

Include dependency graph for interruptions.c:



Functions

- **CY_ISR** (ISR_WATCHDOG_Handler)
- **CY_ISR** (ISR_RS485_RX_ExInterrupt)
- void **interrupt_manager** ()
- void **function_scheduler** ()

5.8.1 Detailed Description

Interruption functions are in this file.

Date

October 01, 2017

Author

Centro "E.Piaggio"

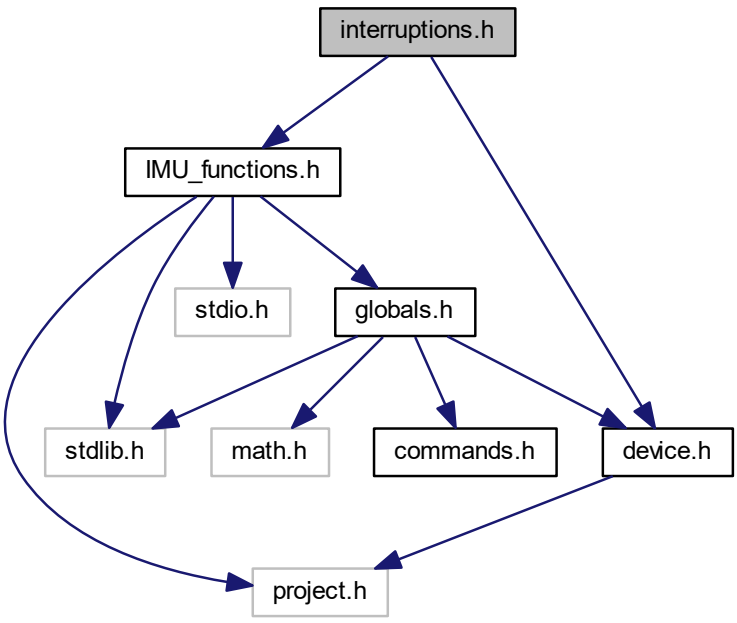
Copyright

5.9 interruptions.h File Reference

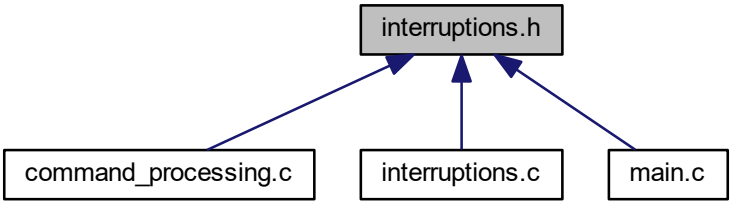
Interruptions header file.

```
#include <device.h>
#include <IMU_functions.h>
```

Include dependency graph for interruptions.h:



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



Functions

- **CY_ISR_PROTO** (ISR_RS485_RX_ExInterrupt)
- **CY_ISR_PROTO** (ISR_WATCHDOG_Handler)
- void **function_scheduler** ()
- void **analog_read_end** ()
- void **interrupt_manager** ()

5.9.1 Detailed Description

Interruptions header file.

Date

October 01, 2017

Author

Centro "E.Piaggio"

Copyright

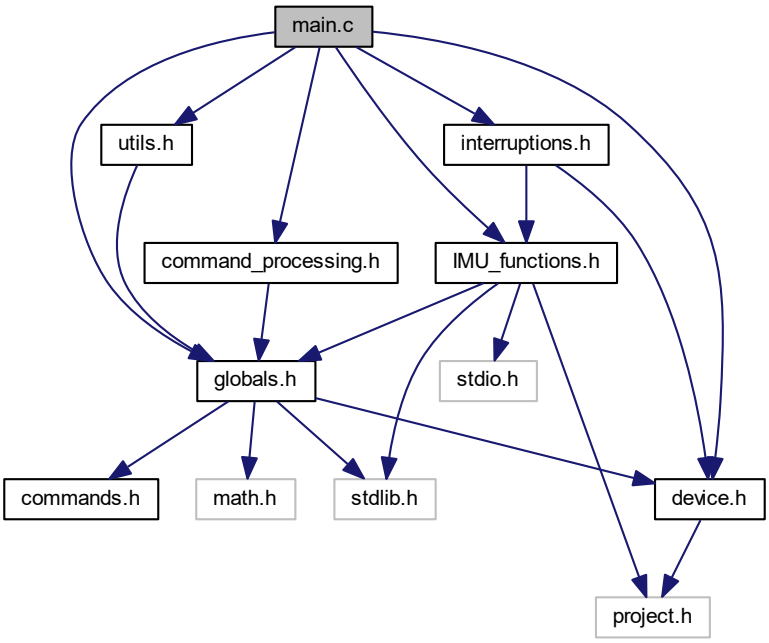
5.10 main.c File Reference

Firmware main file.

```
#include <device.h>
#include <globals.h>
#include <interruptions.h>
#include <command_processing.h>
#include <utils.h>
```

```
#include <IMU_functions.h>
```

Include dependency graph for main.c:



Functions

- int **main** ()

5.10.1 Detailed Description

Firmware main file.

Date

October 01, 2017

Author

Centro "E.Piaggio"

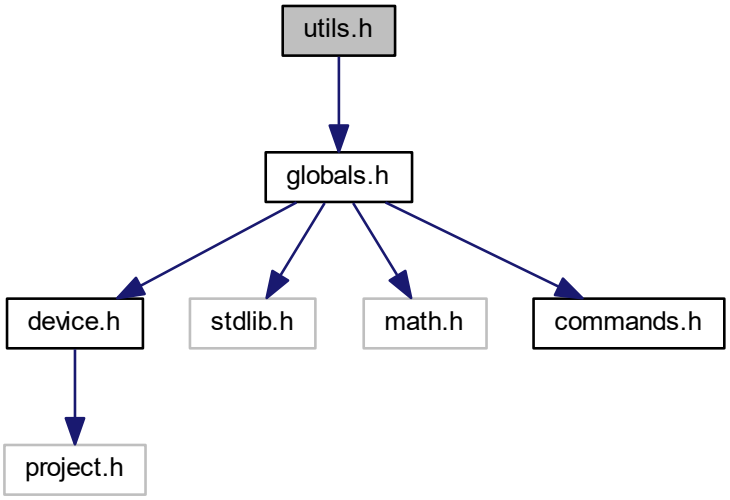
Copyright

5.11 utils.h File Reference

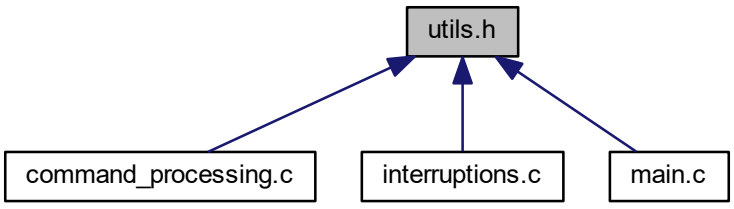
Definition of utility functions.

```
#include <globals.h>
```

Include dependency graph for utils.h:



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



Macros

- #define **TIMER_CLOCK** 10000
- #define **ALPHA** 3
- #define **BETA** 50
- #define **SIGN**(A) (((A) >=0) ? (1) : (-1))

Functions

- int32 **filter_v** (int32 new_value)
- int32 **filter_ch1** (int32 value)
- int32 **filter_ch2** (int32 value)
- int32 **filter_i1** (int32 value)
- int **my_round** (const double x)
- uint32 **my_mod** (int32 val, int32 divisor)
- CYBIT **check_enc_data** (const uint32 *)
- int **calc_turns_fcn** (const int32, const int32)
- void **calibration** ()
- void **torque_feedback** ()

5.11.1 Detailed Description

Definition of utility functions.

Declaration of utility functions.

Date

October 01, 2017

Author

Centro "E.Piaggio"

Copyright

Index