main.c File Reference

: Main program body More...

```
#include "main.h"
#include "Const.h"
#include "stdio.h"
#include <math.h>
#include "function.h"
```

Functions

void	**SystemClock_Config** (void) System Clock Configuration. More...
int	**__io_putchar** (int ch)
void	**Init** (**I2C_HandleTypeDef** *p_hi2c1) Init I2C1 @Note Cette fonction met à 1 le bit 7 de PWR_MGMT_1 pour faire un reset de l'ensemble des registres du capteur, puis attend 100ms et choisit une horlorge. De plus, elle désactive l'interface master I2C du MPU-9250 pour permettre la lecture des données du magnétomètre et elle configure ce dernier pour qu'il réalise des mesures en continu. More...
void	**Measure_T** (**I2C_HandleTypeDef** *hi2cx, double *Temperature) Mesure temperature @Note Cette fonction permet de lire la temperature du capteur puis de l'enregistrer dans une variable globale de type double. More...
void	**Measure_A** (**I2C_HandleTypeDef** *hi2cx, double *Acceleration) Mesure acceleration @Note Cette fonction permet de lire l'acceleration du capteur sur les axes x,y et z puis de l'enregistrer dans une variable globale de type double. More...
void	**Measure_Vitesse_angulaire** (**I2C_HandleTypeDef** *hi2cx, double *tableau_donnee_utiles) Mesure vitesse angulaire @Note Cette fonction permet de lire la vitesse angulaire du capteur sur les axes x,y et z puis de l'enregistrer dans une variable globale de type double. More...
void	**Measure_M** (**I2C_HandleTypeDef** *p_hi2c1, double *mag) Mesure champ magnétique @Note Cette fonction permet de lire la valeur du champ magnétique selon un repère d'axes x,y et z puis de l'enregistrer dans une variable globale de type double. More...
void	**Noise_G** (**I2C_HandleTypeDef** *hi2cx, double *noise) Mesure de la valeur efficace du bruit @Note Cette fonction permet de calculer la valeur efficace du bruit du gyromètre selon une approche probabiliste, en calculant la moyenne d'un échantillon de données de vitesse angulaire. More...
void	**Calib_gyro** (**I2C_HandleTypeDef** *hi2cx, double *gyro_offset) Calibration du gyromètre @Note Cette fonction permet de calculer l'offset des composantes de la vitesse angulaire en se basant sur la méthode du single point. More...
void	**Average** (**I2C_HandleTypeDef** *hi2cx, double *tableau_moyenne) Vérification du fonctionnement de la calibration @Note Cette fonction permet de tester la fonction Cali_gyro en calculant la moyenne de la vitesse angulaire moins son offset lorsque le capteur est immobile. More...
int	**main** (void)

The application entry point. More...

void **Error_Handler** (void)

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

Variables

I2C_HandleTypeDef	**hi2c1**
UART_HandleTypeDef	**hlpuart1**
char	**mess1** [30]
double	**Temperature** = 0
double	**Acceleration** [3] = {0}
double	**norme_vecteur_gravite** = 0
double	**donnees_Gyrometre** [3] ={0}
double	**norme_vecteur_Gyrometre** =0
double	**donnees_mag** [3] ={0}
double	**noise** [3] = {0}
double	**tableau_moyenne** [3] ={0}

Detailed Description

: Main program body

Attention

Function Documentation

◆ Average()

```
void Average ( I2C_HandleTypeDef *  hi2cx,
               double *             tableau_moyenne
             )
```

Vérification du fonctionnement de la calibration @Note Cette fonction permet de tester la fonction Cali_gyro en calculant la moyenne de la vitesse angulaire moins son offset lorsque le capteur est immobile.

Parameters

hi2cx	Pointeur vers une structure I2C qui contient l'information de configuration pour un i2c particulier
tableau_moyenne	Pointeur vers une zone mémoire de type double contenant l'information de valeur moyenne de la vitesse angulaire calibrée et capteur immobile (censée valoir 0)

Return values

None

◆ Calib_gyro()

```
void Calib_gyro ( I2C_HandleTypeDef *  hi2cx,
                  double *             gyro_offset
                )
```

Calibration du gyromètre @Note Cette fonction permet de calculer l'offset des composantes de la vitesse angulaire en se basant sur la méthode du single point.

Parameters

hi2cx	Pointeur vers une structure I2C qui contient l'information de configuration pour un i2c particulier
gyro_offset	Pointeur vers une zone mémoire de type double contenant l'information de l'offset du gyromètre

Return values

None

◆ Error_Handler()

void Error_Handler (void)

This function is executed in case of error occurrence.

Return values

None

◆ Init()

void Init (**I2C_HandleTypeDef** * p_hi2c1)

Init I2C1 @Note Cette fonction met à 1 le bit 7 de PWR_MGMT_1 pour faire un reset de l'ensemble des registres du capteur, puis attend 100ms et choisit une horlorge. De plus, elle désactive l'interface master I2C du MPU-9250 pour permettre la lecture des données du magnétomètre et elle configure ce dernier pour qu'il réalise des mesures en continu.

Parameters

p_hi2c1 Pointeur vers une structure I2C qui contient l'information de configuration pour un i2c particulier

Return values

None

◆ main()

int main (void)

The application entry point.

Return values

int

◆ Measure_A()

```
void Measure_A ( I2C_HandleTypeDef *  hi2cx,
                 double *             Acceleration
               )
```

Mesure acceleration @Note Cette fonction permet de lire l'acceleration du capteur sur les axes x,y et z puis de l'enregistrer dans une variable globale de type double.

Parameters

hi2cx	Pointeur vers une structure I2C qui contient l'information de configuration pour un i2c particulier
Acceleration	Pointeur vers une zone mémoire de type double contenant l'information d'accélération

Return values

None

◆ Measure_M()

```
void Measure_M ( I2C_HandleTypeDef *  p_hi2c1,
                 double *             mag
               )
```

Mesure champ magnétique @Note Cette fonction permet de lire la valeur du champ magnétique selon un repère d'axes x,y et z puis de l'enregistrer dans une variable globale de type double.

Parameters

hi2cx	Pointeur vers une structure I2C qui contient l'information de configuration pour un i2c particulier
tableau_donnee_utiles	Pointeur vers une zone mémoire de type double contenant l'information de champ magnétique

Return values

None

◆ Measure_T()

```
void Measure_T ( I2C_HandleTypeDef *  hi2cx,
                 double *             Temperature
               )
```

Mesure temperature @Note Cette fonction permet de lire la temperature du capteur puis de l'enregistrer dans une variable globale de type double.

Parameters

hi2cx	Pointeur vers une structure I2C qui contient l'information de configuration pour un i2c particulier.
Temperature	Pointeur vers une zone mémoire de type double contenant l'information de température

Return values

None

◆ Measure_Vitesse_angulaire()

```
void Measure_Vitesse_angulaire ( I2C_HandleTypeDef *  hi2cx,
                                 double *             tableau_donnee_utiles
                               )
```

Mesure vitesse angulaire @Note Cette fonction permet de lire la vitesse angulaire du capteur sur les axes x,y et z puis de l'enregistrer dans une variable globale de type double.

Parameters

hi2cx	Pointeur vers une structure I2C qui contient l'information de configuration pour un i2c particulier
tableau_donnee_utiles	Pointeur vers une zone mémoire de type double contenant l'information de vitesse angulaire

Return values

None

◆ Noise_G()

void Noise_G (**I2C_HandleTypeDef** * hi2cx,
double * noise
)

Mesure de la valeur efficace du bruit @Note Cette fonction permet de calculer la valeur efficace du bruit du gyromètre selon une approche probabiliste, en calculant la moyenne d'un échantillon de données de vitesse angulaire.

Parameters

hi2cx Pointeur vers une structure I2C qui contient l'information de configuration pour un i2c particulier

noise Pointeur vers une zone mémoire de type double contenant l'information de valeur efficace du bruit

Return values

None

◆ 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.

Initializes the CPU, AHB and APB buses clocks