

# HLV\_Configuracion\_Inicial

## 1.0

Generated by Doxygen 1.9.8



---

<b>1 Class Index</b>	<b>1</b>
1.1 Class List . . . . .	1
<b>2 File Index</b>	<b>3</b>
2.1 File List . . . . .	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 Float_Byte Union Reference . . . . .	5
3.1.1 Detailed Description . . . . .	5
3.1.2 Member Data Documentation . . . . .	5
3.1.2.1 dato . . . . .	5
3.1.2.2 f . . . . .	5
<b>4 File Documentation</b>	<b>7</b>
4.1 main.c File Reference . . . . .	7
4.1.1 Typedef Documentation . . . . .	7
4.1.1.1 Flotante . . . . .	7
4.1.2 Function Documentation . . . . .	8
4.1.2.1 calibrate_sensor() . . . . .	8
4.1.2.2 config_Sensors() . . . . .	8
4.1.2.3 erase_EEPROM() . . . . .	9
4.1.2.4 main() . . . . .	9
<b>Index</b>	<b>11</b>



# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Float_Byte</a>	Declaracion de un tipo de dato union para la separacion de flotantes en Bytes . . . . .	<a href="#">5</a>
----------------------------	---	-------------------



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">main.c</a> . . . . .	7
----------------------------------	---





## Chapter 3

# Class Documentation

### 3.1 Float\_Byte Union Reference

Declaracion de un tipo de dato union para la separacion de flotantes en Bytes.

#### Public Attributes

- float [f](#)
- unsigned char [dato](#) [4]

#### 3.1.1 Detailed Description

Declaracion de un tipo de dato union para la separacion de flotantes en Bytes.

#### 3.1.2 Member Data Documentation

##### 3.1.2.1 dato

```
unsigned char Float_Byte::dato[4]
```

##### 3.1.2.2 f

```
float Float_Byte::f
```

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

- [main.c](#)



# Chapter 4

## File Documentation

### 4.1 main.c File Reference

```
#include <avr/io.h>
```

#### Classes

- union [Float\\_Byte](#)

*Declaracion de un tipo de dato union para la separacion de flotantes en Bytes.*

#### Typedefs

- typedef union [Float\\_Byte](#) [Flotante](#)

*Declaracion de un tipo de dato union para la separacion de flotantes en Bytes.*

#### Functions

- void [erase\\_EEPROM](#) ()

*Borra todos los registros de la data EEPROM.*

- void [config\\_Sensors](#) (char \*S1, char \*S2, char \*S3)

*Configura los posibles sensores de la HLV-Attiny1626.*

- void [calibrate\\_sensor](#) (float a, float b, int port)

*Configura las posibles rectas de calibracion para los sensores 4.20ma.*

- int [main](#) ()

*Funcion principal para el manejo de registros.*

#### 4.1.1 Typedef Documentation

##### 4.1.1.1 Flotante

```
typedef union Float\_Byte Flotante
```

Declaracion de un tipo de dato union para la separacion de flotantes en Bytes.

## 4.1.2 Function Documentation

### 4.1.2.1 `calibrate_sensor()`

```
void calibrate_sensor (
    float a,
    float b,
    int port )
```

Configura las posibles rectas de calibracion para los sensores 4.20ma.

#### Parameters

<i>a</i>	Dato del tipo flotante que corresponde al parametro a en la ecuacion $ax+b$
<i>b</i>	Dato del tipo flotante que corresponde al parametro en la ecuacion $ax+b$
<i>port</i>	Dato del tipo entero que indica la posicion del sensor a calibrar

### 4.1.2.2 `config_Sensors()`

```
void config_Sensors (
    char * S1,
    char * S2,
    char * S3 )
```

Configura los posibles sensores de la HLV-Attiny1626.

#### Parameters

<i>S1</i>	Caracter de configuracion del sensor 1
-----------	--

Para el sensor 1 existen las posibilidades:

```
P: Sensor 4.20mA
I: Sensor I2C 20bar
i: Sensor I2C 7bar
N: Sensor 4.20mA de sumergimiento
```

#### Parameters

<i>S2</i>	Caracter de configuracion del sensor 2
-----------	--

Para el sensor 2 existen las posibilidades:

```
P: Sensor 4.20mA
I: Sensor I2C 20bar
```

i: Sensor I2C 7bar  
N: Sensor 4.20mA de sumergimiento

#### Parameters

S3	Caracter de configuracion del sensor 3
----	--

Para el sensor 3 existen las posibilidades de flujo:

0 = 1LPS  
1 = 10LPS  
2 = 100LPS  
3 = 1000LPS  
4 = 10000LPS

#### 4.1.2.3 erase\_EEPROM()

```
void erase_EEPROM ( )
```

Borra todos los registros de la data EEPROM.

#### 4.1.2.4 main()

```
int main ( )
```

Funcion principal para el manejo de registros.



# Index

- calibrate\_sensor
  - main.c, [8](#)
- config\_Sensors
  - main.c, [8](#)
- dato
  - Float\_Byte, [5](#)
- erase\_EEPROM
  - main.c, [9](#)
- f
  - Float\_Byte, [5](#)
- Float\_Byte, [5](#)
  - dato, [5](#)
  - f, [5](#)
- Flotante
  - main.c, [7](#)
- main
  - main.c, [9](#)
- main.c, [7](#)
  - calibrate\_sensor, [8](#)
  - config\_Sensors, [8](#)
  - erase\_EEPROM, [9](#)
  - Flotante, [7](#)
  - main, [9](#)