IsoTemp: Arduino

Contents

[Configuración 1](#_Toc422755406)

[Defines 1](#_Toc422755407)

[Archivo IsoTemp.h 2](#_Toc422755408)

[Protocolo de serialización 3](#_Toc422755409)

[Especificación de cada campo 3](#_Toc422755410)

[Representacion del protocol mediante defines 3](#_Toc422755411)

# Configuración

Para configurar el programa IsoTemp se debe modificar el archivo IsoTemp.h modificando defines. Se lo hace de esta manera ya que Arduino por si mismo no puede leer o escribir archivos, a menos que se instale un hardware SD Card.

## Defines

**#define BAUD\_RATE 9600**

Permite setear la velocidad del Puerto serie. Este deberá coincidir con el baud rate en la aplicación del servidor.

**#define TEST**

Permite configurar la aplicación para ejecute en modo test, simplemente dejando la línea sin comentar.

Mediante el modo test la aplicación enviara por serial información aleatoria de los sensores.

Para configurar el modo Normal se tiene que comentar la línea, quedando de la siguiente manera:

//#define TEST

En este modo, la información enviada al puerto serie se tomara de los sensores.

**#define NAME\_TEMPERATURE\_SENSOR "Temperatura"**

**#define NAME\_HUMIDITY\_SENSOR "Humedad"**

**#define NAME\_GAS\_SENSOR "Gas"**

Mediante estas definiciones se configura el nombre del sensor. Dicho nombre se incluirá en la información serializada por el puerto.

**#define UNIT\_TEMPERATURE "\*C"**

**#define UNIT\_HUMIDITY "%"**

**#define UNIT\_GAS "?"**

Mediante estas definiciones se configura la unidad de medida de la información que devuelve cada uno de los sensores. Dicha unidad se incluirá en la información serializada por el puerto.

**#define SPACE\_TOKEN "\t\t"**

Esta definición se utiliza para separar la información de cada uno de los sensores

**#define DELAY 2000**

Este delay es utilizado para establecer el tiempo entre cada adquisición de datos de los sensores.

**#define MQ2PIN A0**

MQ2PIN Indica el pin para el sensor MQ2, utilizado para medir composición de gases.

**#define DHTPIN 8**

DHTPIN Indica el pin para el sensor DHTxx, utilizado para medir temperatura.

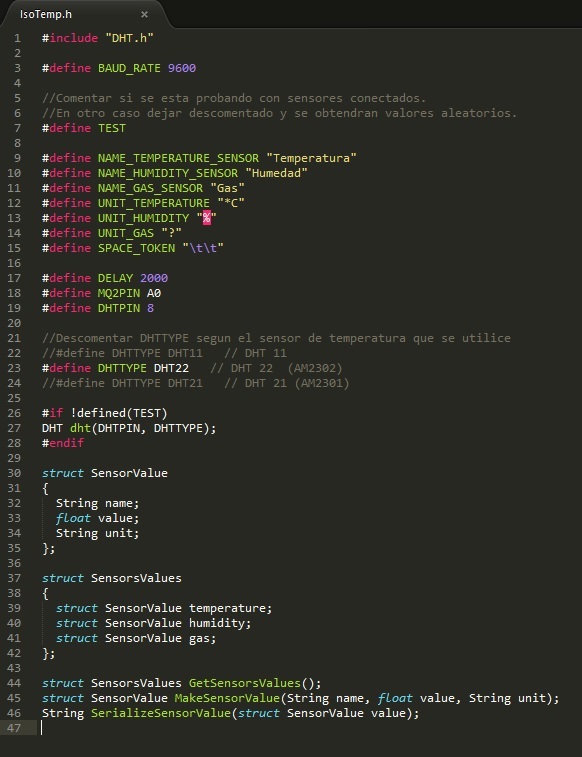
**//#define DHTTYPE DHT11 // DHT 11**

**#define DHTTYPE DHT22 // DHT 22 (AM2302)**

**//#define DHTTYPE DHT21 // DHT 21 (AM2301)**

Estas definiciones se utilizan para inicializar el sensor de temperature, indicando el tipo de sensor. Por lo cual se deberá descomentar solamente un define dependiendo del sensor utilizado.

# Archivo IsoTemp.h



# 

# Protocolo de serialización

Línea de texto que se envía por serial (Serial.println(“Texto”)).

SENSORS\_DATA\_LINE = SENSOR\_INFORMATION + SPACE\_TOKEN + SENSOR\_INFORMATION + SPACE\_TOKEN + SENSOR\_INFORMATION

## Especificación de cada campo

SENSOR\_INFORMATION = <NAME\_SENSOR: FLOAT\_NUMBER [UNIT]>

NAME\_SENSOR = Cadena string que representa nombre del sensor.

FLOAT\_NUMBER = Número flotante que representa el dato obtenido por el sensor.

UNIT = Cadena string que representa la unidad de medida de los datos de un sensor.

SPACE\_TOKEN = string que separa la información de cada sensor (SENSOR\_INFORMATION)

## Representacion del protocol mediante defines

<NAME\_GAS\_SENSOR: FLOAT\_NUMBER [UNIT\_GAS]>SPACE\_TOKEN<NAME\_TEMPERATURE\_SENSOR: FLOAT\_NUMBER [UNIT\_TEMPERATURE]> SPACE\_TOKEN <NAME\_HUMIDITY\_SENSOR: FLOAT\_NUMBER [UNIT\_HUMIDITY]>

FLOAT\_NUMBER es un número float y no está definido en un #define.

Ejemplos:

<Gas: 23.00 [Unit]> <Temperature: 34.00 [Unit]> <Humidity: 78.00 [Unit]>

<Gas: NaN [Unit]> <Temperature: NaN [Unit]> <Humidity: NaN [Unit]>

En caso de que haya algún error de lectura se utilizara el símbolo Nan(Not a number) para indicar un error en la adquisición del sensor.