

CURSO DE ARDUINO

DIRIGIDO POR: MIGUEL ANGEL CALIFA URQUIZA



AGENDA

- Buenas practicas a la hora de elaborar un programa en Arduino.
- Objetos en Arduino, clases, librerías.
- Ejemplos con librerías.

BUENAS PRACTICAS A LA HORA DE HACER UN PROGRAMA EN ARDUINO

- Usar componentes propios del lenguaje C en Arduino, ejemplo:

```
#define nombreConstante valor
```

Tenga en cuenta que # es necesario.

```
#define led 13
```

Nota:

No hay punto y coma después de la instrucción #define. Si se incluye uno, el compilador genera errores crípticos más abajo en la página.

SEPARAR EL CÓDIGO EN OBJETOS (.H, .CPP)

```
1  #include "SERIAL_COMMUNICATION.h"
2  #include "PROCESS_DATA.h"
3  #include "WIFI_PROCESS.h"
4  #include "MEMORY_ADMINISTRATION.h"
5  #include "PINS.h"
6  #include "onmotica.h"
7  #include "configuration.h"
8
```

```
19  SERIAL_COMMUNICATION serial;
20  PROCESS_DATA procesamiento;
21  WIFI_PROCESS WiFiProcess;
22  MEMORY_ADMINISTRATION administracion;
23  PINS pinesIO;
24  onmotica utils;
25  DHT dht(DHTPIN, DHTTYPE);
26
```

Branch: master On_TempHum_logger / DL_Onmotica / configuration.h Find file Copy path

49965412 Modified 4 days ago 49965418 17 days ago

Contributors

44 lines (16 sloc) 1.49 KB

```
1  #ifndef CONFIGURATION_H
2  #define CONFIGURATION_H
3
4  #include "Arduino.h"
5
6  #define serDebug true //Para activar o desactivar el debug serial
7  #define BAUD_RATE 115200
8
9  //Mosquitto
10 //Server ip-addr onmotica and public bypass servers
11 //Info: https://github.com/ajdt/mqtt.github.io/wiki/public_brokers
12 //static const char* mqtt_server[] = {"157.140.174.83", "test.mosquitto.org", "iot.eclipse.org", "broker.hivemq.com", "www.cloudmqtt.com",
13
14 #define outTopic "/Demo/Cens/Output"
15 #define inTopic "/Demo/Cens/Input"
16 #define ulpiname "Cens_Cuota_Inf_Module_1"
17
18 #define ledWarning led_builtin
19 #define chipSelect D4
20 #define arduinoTSPin D5
21 #define arduinoRSPin D2
22 #define arduinoResetPin D1 //pin que comunica con el reset del arduino
23
24 #define timeDelay 5*60*1000 // 5 minutos * 60 segundos * 1000 milisegundos
25 #define minTime 100 //Tiempo minimo para re-enviar una actualizacion milisegundos
26 #define tabulador "," //Item tabulador
27 #define minDelay 30 //delay min for try
28
29 #define maxTimeUntilPublish 30*60*1000 //30 min max de 30 minutos sin publicar es preocupante
30 #define maxError 500 //Numero maximo de errores tolerable antes de causar un reinicio del arduino
31
```

USAR CONSTANTES DONDE SEA POSIBLE

```
const float pi = 3.14;
```

```
float x;
```

```
// ....
```

```
x = pi * 2;  //uso correcto de la constante en matemáticas
```

```
pi = 7;      // ilegal - no se puede escribir (modificar) una constante
```


LIBRERIAS EN ARDUINO

- Las librerías proporcionan funciones adicionales que se pueden usar en un programa.

Hay dos tipos de librerías:

- * Librerías oficiales (Arduino)
- * Librerías hechas por la comunidad

- [EEPROM](#) : lectura y escritura en almacenamiento "permanente"
- [Ethernet](#) : para conectarse a Internet con Arduino Ethernet Shield, Arduino Ethernet Shield 2 y Arduino Leonardo ETH
- [Firmata](#) : para comunicarse con aplicaciones en la computadora mediante un protocolo serie estándar.
- [GSM](#) : para conectarse a una red GSM / GRPS con el escudo GSM.
- [LiquidCrystal](#) - para controlar pantallas de cristal líquido (LCD)
- [SD](#) : para leer y escribir tarjetas SD
- [Servo](#) - para controlar servomotores
- [SPI](#) : para comunicarse con dispositivos que utilizan el bus de interfaz periférica serie (SPI)
- [Software Serial](#) : para comunicación en serie en cualquier pin digital. La versión 1.0 y posteriores de Arduino incorporan la biblioteca NewSoftSerial de [Mikal Hart](#) como SoftwareSerial .
- [Paso a paso](#) : para controlar motores paso a paso
- [TFT](#) : para dibujar texto, imágenes y formas en la pantalla Arduino TFT
- [WiFi](#) : para conectarse a Internet utilizando el escudo Arduino WiFi
- [Cable](#) : interfaz de dos cables (TWI / I2C) para enviar y recibir datos a través de una red de dispositivos o sensores.

Las bibliotecas Matrix y Sprite ya no son parte de la distribución principal.

COMPONENTES DE UNA LIBRERIA

1. Código fuente.
2. Ejemplos.
3. Licencia.
4. Library.properties.

Blink.h:

```
1  /*
2  blink.h - Simple example in creating your own Arduino Library
3  Copyright (c) 2017 op of TMM. All right reserved.
4
5  A pin is blinked automatically by one second intervals or by a speci
6  Methods:
7
8  Blink(pin) - Constructor. Specify pin to blink
9  blink(value) - Enable blinking. Accepts 1, 0 or ON and OFF
10 blink(value, length) - Enable blinking and specify interval of blink
11 */
12
13 #ifndef Blink_h
14 #define Blink_h
15
16 #include <Arduino.h>
17
18 #define ON true
19 #define OFF false
20
21 class Blink
22 {
23 public:
24   Blink(int pin); //Constructor. attach pin to blink
25   void blink(bool value); //enable blinking with is duration
26   void blink(bool value, int blinkLength); //enable blinking with bl
27   void blink(bool value, int blinkLength, int blinkLoops); //enable b
28 private:
29   uint8_t pinNumber;
30 };
31
32 #endif
```

Blink.cpp

```
1  /*
2  blink.cpp - Simple example in creating your own Arduino Library
3  Copyright (c) 2017 op of TMM. All right reserved.
4
5  */
6
7  #include <Arduino.h>
8  #include "Blink.h"
9
10 Blink::Blink(int pin){
11   pinMode(pin, OUTPUT);
12   pinNumber = pin;
13 }
14
15 void Blink::blink(bool value){
16   if(value == true){
17     digitalWrite(pinNumber,HIGH);
18     delay(1000);
19     digitalWrite(pinNumber,LOW);
20     delay(1000);
21   }else{
22     digitalWrite(pinNumber,LOW);
23   }
24 }
25
26 void Blink::blink(bool value, int blinkLength){
27   if(value == true){
28     digitalWrite(pinNumber,HIGH);
29     delay(blinkLength);
30     digitalWrite(pinNumber,LOW);
31     delay(blinkLength);
32   }else{
33     digitalWrite(pinNumber,LOW);
34   }
35 }
36
37 void Blink::blink(bool value, int blinkLength, int blinkLoops){
38   if(value == true){
39     for(int i=0;i<blinkLoops;i++){
40       digitalWrite(pinNumber,HIGH);
41       delay(blinkLength);
42       digitalWrite(pinNumber,LOW);
43       delay(blinkLength);
44     }
45   }else{
46     digitalWrite(pinNumber,LOW);
47   }
48 }
```

ESTRUCTURA DE FOLDERS AL INTERIOR DE UNA LIBRERÍA (RECOMENDACIÓN)

Cada librería debería mínimo contener los siguientes folders:

src = En donde esta el código de la librería (.h, .cpp)

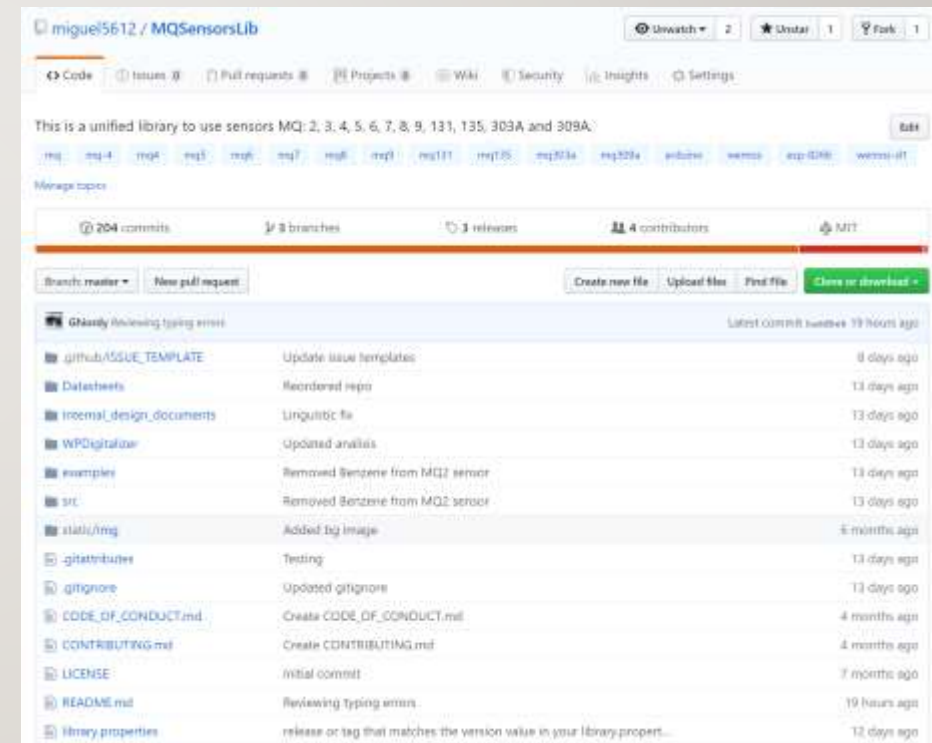
Examples = Espacio donde están los ejemplos

Y en el raíz del repositorio:

LICENSE

Library.properties

README.md



COMO SE CODIFICA UNA LIBRERÍA (SRC/*)

Blink.h:

```
1 /*
2  blink.h - Simple example in creating your own Arduino Library
3  Copyright (c) 2017 op of TMM. All right reserved.
4
5  A pin is blinked automatically by one second intervals or by a spect
6  Methods:
7
8  Blink(pin) - Constructor. Specify pin to blink
9  blink(value) - Enable blinking. Accepts 1, 0 or ON and OFF
10 blink(value, length) - Enable blinking and specify interval of blink
11 */
12
13 #ifndef Blink_h
14 #define Blink_h
15
16 #include <Arduino.h>
17
18 #define ON true
19 #define OFF false
20
21 class Blink
22 {
23 public:
24   Blink(int pin); //Constructor. attach pin to blink
25   void blink(bool value); //enable blinking with 1s duration
26   void blink(bool value, int blinkLength); //enable blinking with bl
27   void blink(bool value, int blinkLength, int blinkLoops); //enable b
28 private:
29   uint8_t pinNumber;
30 };
31
32 #endif
```

Blink.cpp

```
1 /*
2  blink.cpp - Simple example in creating your own Arduino Library
3  Copyright (c) op from TMM. All right reserved.
4
5  */
6
7 #include <Arduino.h>
8 #include "Blink.h"
9
10 Blink::Blink(int pin){
11   pinMode(pin, OUTPUT);
12   pinNumber = pin;
13 }
14
15 void Blink::blink(bool value){
16   if(value == true){
17     digitalWrite(pinNumber,HIGH);
18     delay(1000);
19     digitalWrite(pinNumber,LOW);
20     delay(1000);
21   }else{
22     digitalWrite(pinNumber,LOW);
23   }
24 }
25
26 void Blink::blink(bool value, int blinkLength ){
27   if(value == true){
28     digitalWrite(pinNumber,HIGH);
29     delay(blinkLength);
30     digitalWrite(pinNumber,LOW);
31     delay(blinkLength);
32   }else{
33     digitalWrite(pinNumber,LOW);
34   }
35 }
36
37 void Blink::blink(bool value, int blinkLength, int blinkLoops){
38   if(value == true){
39     for(int i=0;i<blinkLoops;i++){
40       digitalWrite(pinNumber,HIGH);
41       delay(blinkLength);
42       digitalWrite(pinNumber,LOW);
43       delay(blinkLength);
44     }
45   }else{
46     digitalWrite(pinNumber,LOW);
47   }
48 }
```

COMO SE IMPORTA LA LIBRERIA

1. Crear un archivo .ZIP que contenga los folders de la librería.
2. Usando el menú programa/Incluir Librería/Agregar Libreria (.zip).
3. Usar la librería mediante herramientas/librería/ Hacer clic en la librería como se muestra en la imagen.



EJEMPLO DHT11

```
DHT dht(DHTPIN, DHTTYPE);

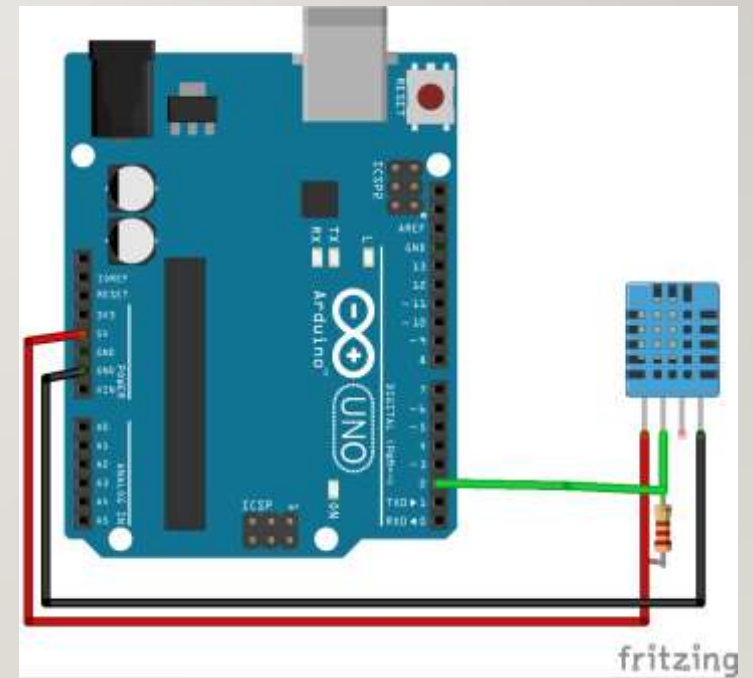
void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");

  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
}
```



EJEMPLO MQ-2

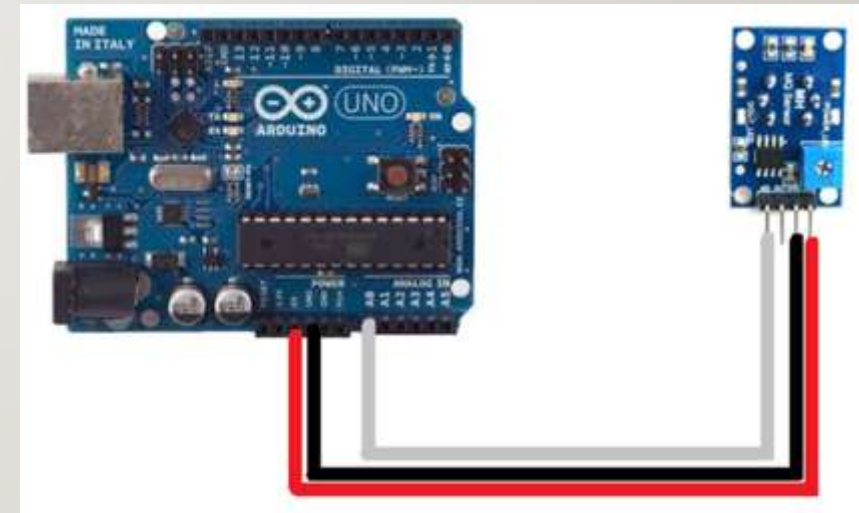
```
#define pin A0 //Analog input 0 of your arduino
#define type 2 //MQ2
//define calibration_button 13 //Pin to calibrate your sensor

//Declare Sensor
MQUnifiedSensor MQ2(pin, type);

//Variables
float H2, LPG, CO, Alcohol, Propane, Benzene;

void setup() {
  Serial.begin(9600); //Init serial port
  /***** MQUnifiedSensor *****/
  Input:
  Output:
  Remarks: This function configure the pinMode
  /*****
  //Init the sensor
  MQ2.initialise();
  //pinMode(calibration_button, INPUT);
  */
}

void loop() {
  MQ2.update(); // Update data, the arduino will be read the voltage in the analog pin
  /*
  //Routine de calibration - Uncomment if you need (setup two and header)
  if(calibration_button)
  {
    float H2 = MQ2.calibrate();
    MQ2.setH2(H2);
  }
  */
  /***** MQUnifiedSensor *****/
  Input: Gas - serial print flag
  Output: Value in ppm
  Remarks: This function use readPPM to read the value in ppm the gas in the air.
  /*****
  //Read the sensor and print in serial port
  //lecture will be saved in lecture Variable
  //float lecture = MQ2.readSensor("", true); // Return LPG concentration
  // Options, uncomment where you need
  H2 = MQ2.readSensor("H2"); // Return H2 concentration
  LPG = MQ2.readSensor("LPG"); // Return LPG concentration
  CO = MQ2.readSensor("CO"); // Return CO concentration
  */
}
```



EJEMPLO CCS-811

```
CCS811_test

#include "Adafruit_CCS811.h"

Adafruit_CCS811 ccs;

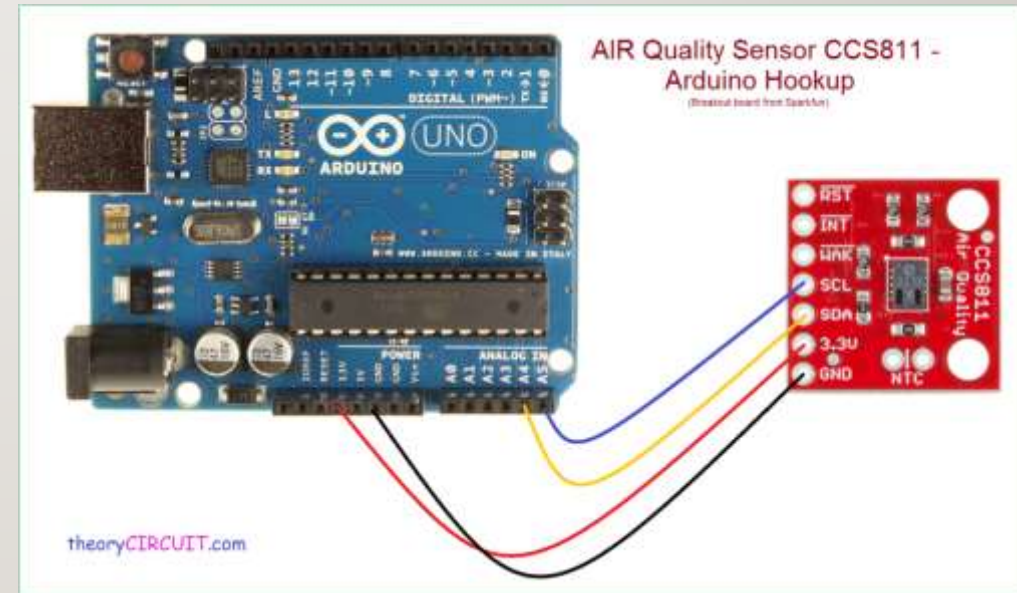
void setup() {
  Serial.begin(9600);

  Serial.println("CCS811 test");

  if(!ccs.begin()){
    Serial.println("Failed to start sensor! Please check your wiring.");
    while(1);
  }

  //calibrate temperature sensor
  while(!ccs.available());
  float temp = ccs.calculateTemperature();
  ccs.setTempOffset(temp - 25.0);
}

void loop() {
  if(ccs.available()){
    float temp = ccs.calculateTemperature();
    if(!ccs.readData()){
      Serial.print("CO2: ");
      Serial.print(ccs.getCO2());
      Serial.print("ppm, TVOC: ");
      Serial.print(ccs.getTVOC());
      Serial.print("ppb Temp:");
      Serial.println(temp);
    }
  }
}
```



PREGUNTAS



TRABAJO INVESTIGATIVO

- Implemente un sistema de luces rítmicas enviando comandos por puerto serie.
- Verifique que el código fuente haya seguido las buenas practicas mencionadas en clase.
- Proponga un proyecto para resolver un problema en su hogar/empresa usando arduino.