

Práctica 5 Display:

Código:

```
#include <Arduino.h>

void init_temp_hum_task(void);

// For a connection via I2C using the Arduino Wire include:
#include <Wire.h>           // Only needed for Arduino 1.6.5 and earlier
#include "SSD1306Wire.h"    // legacy: #include "SSD1306.h"
// OR #include "SH1106Wire.h" // legacy: #include "SH1106.h"

// For a connection via I2C using brzo_i2c (must be installed) include:
// #include <brzo_i2c.h>      // Only needed for Arduino 1.6.5 and earlier
// #include "SSD1306Brzo.h"
// OR #include "SH1106Brzo.h"

// For a connection via SPI include:
// #include <SPI.h>           // Only needed for Arduino 1.6.5 and earlier
// #include "SSD1306Spi.h"
// OR #include "SH1106Spi.h"

// Optionally include custom images
#include "images.h"

#include "SparkFunHTU21D.h"

//Create an instance of the object
HTU21D myHumidity;

// Initialize the OLED display using Arduino Wire:
SSD1306Wire display(0x3c, SDA, SCL); // ADDRESS, SDA, SCL - SDA and SCL usually populate automatically based on your board's pin
// SSD1306Wire display(0x3c, D3, D5); // ADDRESS, SDA, SCL - If not, they can be specified manually.
// SSD1306Wire display(0x3c, SDA, SCL, GEOMETRY_128_32); // ADDRESS, SDA, SCL, OLEDDISPLAY_GEOMETRY - Extra param required for 12
// SH1106Wire display(0x3c, SDA, SCL); // ADDRESS, SDA, SCL

// Initialize the OLED display using brzo_i2c:
// SSD1306Brzo display(0x3c, D3, D5); // ADDRESS, SDA, SCL
// or
// SH1106Brzo display(0x3c, D3, D5); // ADDRESS, SDA, SCL

// Initialize the OLED display using SPI:
// D5 -> CLK
// D7 -> MOSI (DOUT)
// D0 -> RES
// D2 -> DC
// D8 -> CS
// SSD1306Spi display(D0, D2, D8); // RES, DC, CS
// or
// SH1106Spi display(D0, D2); // RES, DC

#define DEMO_DURATION 3000
typedef void (*Demo)(void);

int demoMode = 0;
int counter = 1;

void setup() {
    Serial.begin(115200);
```

```

Serial.println();
Serial.println();

    Serial.println("HTU21D Example!");

    myHumidity.begin();
//}

//init_temp_hum_task();
// Initialising the UI will init the display too.
display.init();

display.flipScreenVertically();
display.setFont(ArialMT_Plain_10);

}

void drawFontFaceDemo() {
    // Font Demo1
    // create more fonts at http://oleddisplay.squix.ch/
    display.setTextAlignment(TEXT_ALIGN_LEFT);
    display.setFont(ArialMT_Plain_10);
    display.drawString(0, 0, "Hello world");
    display.setFont(ArialMT_Plain_16);
    display.drawString(0, 10, "Hello world");
    display.setFont(ArialMT_Plain_24);
    display.drawString(0, 26, "Hello world");
}

void drawTextFlowDemo() {
    display.setFont(ArialMT_Plain_10);
    display.setTextAlignment(TEXT_ALIGN_LEFT);
    display.drawStringMaxWidth(0, 0, 128,
        "Lorem ipsum\n dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut l
    )
}

void drawTextAlignmentDemo() {
    // Text alignment demo
    display.setFont(ArialMT_Plain_10);

    // The coordinates define the left starting point of the text
    display.setTextAlignment(TEXT_ALIGN_LEFT);
    display.drawString(0, 10, "Left aligned (0,10)");

    // The coordinates define the center of the text
    display.setTextAlignment(TEXT_ALIGN_CENTER);
    display.drawString(64, 22, "Center aligned (64,22)");

    // The coordinates define the right end of the text
    display.setTextAlignment(TEXT_ALIGN_RIGHT);
    display.drawString(128, 33, "Right aligned (128,33)");
}

void drawRectDemo() {
    // Draw a pixel at given position
    for (int i = 0; i < 10; i++) {
        display.setPixel(i, i);
        display.setPixel(10 - i, i);
    }
    display.drawRect(12, 12, 20, 20);

    // Fill the rectangle
    display.fillRect(14, 14, 17, 17);

    // Draw a line horizontally

```

```

display.drawHorizontalLine(0, 40, 20);

// Draw a line horizontally
display.drawVerticalLine(40, 0, 20);
}

void drawCircleDemo() {
  for (int i = 1; i < 8; i++) {
    display.setColor(WHITE);
    display.drawCircle(32, 32, i * 3);
    if (i % 2 == 0) {
      display.setColor(BLACK);
    }
    display.fillCircle(96, 32, 32 - i * 3);
  }
}

void drawProgressBarDemo() {
  int progress = (counter / 5) % 100;
  // draw the progress bar
  display.drawProgressBar(0, 32, 120, 10, progress);

  // draw the percentage as String
  display.setTextAlignment(TEXT_ALIGN_CENTER);
  display.drawString(64, 15, String(progress) + "%");
}

void drawImageDemo() {
  // see http://blog.squix.org/2015/05/esp8266-nodemcu-how-to-create-xbm.html
  // on how to create xbm files
  display.drawXbm(34, 14, WiFi_Logo_width, WiFi_Logo_height, WiFi_Logo_bits);
}

Demo demos[] = {drawFontFaceDemo, drawTextFlowDemo, drawTextAlignmentDemo, drawRectDemo, drawCircleDemo, drawProgressBarDemo, drawImageDemo};
int demoLength = (sizeof(demos) / sizeof(Demo));
long timeSinceLastModeSwitch = 0;

void loop() {

  float humd = myHumidity.readHumidity();
  float temp = myHumidity.readTemperature();

  Serial.print("Time:");
  Serial.print(millis());
  Serial.print(" Temperature:");
  Serial.print(temp, 1);
  Serial.print("C");
  Serial.print(" Humidity:");
  Serial.print(humd, 1);
  Serial.print("%");

  Serial.println();

  // clear the display
  display.clear();
  // draw the current demo method
  //demos[demoMode]();

  display.setTextAlignment(TEXT_ALIGN_CENTER);
  display.setFont(ArialMT_Plain_10);
  display.drawString(128/2, 0, "HUMEDAD");
  display.setFont(ArialMT_Plain_16);
  display.drawString(128/2, 11, String(humd)+ "%");
  display.setFont(ArialMT_Plain_10);

```

```
display.fillRect(128/2, 30, 128, 30);
display.drawString(128/2, 30, "TEMPERATURA");
display.setFont(ArialMT_Plain_16);
display.drawString(128/2, 41, String(temp)+ "°C");

display.setFont(ArialMT_Plain_10);
display.setTextAlignment(TEXT_ALIGN_RIGHT);
display.drawString(128, 54, String(millis()/3600000)+String(":")\
+String((millis()/60000)%60)+String(":")\
+String((millis()/1000)%60));

// write the buffer to the display
display.display();

delay(100);
}
```

Salida:

Una vez subido el código a la ESP32, el dispositivo, muestra dos cosas:

Pantalla:

Por la pantalla del ordenador muestra el tiempo que lleva en funcionamiento, la lectura de la temperatura actual y la lectura de la humedad actual, las cuales son captadas usando un sensor de temperatura y humedad.

Display:

Por el display muestra ambas lecturas de temperatura y humedad de manera cómoda y organizada como se ve en el siguiente video.

<https://user-images.githubusercontent.com/100867309/171381156-48ad0ec6-ec2e-45cd-b8c0-61bd8b0ca546.mp4>

Montaje:

□

Aclaraciones del código:

Para las aclaraciones vamos a separar el código en varios apartados de manera que sea más fácil de comprender, para empezar tenemos que definir todas las librerías que sean necesarias, definir variables, etc... es decir hay que inicializar el código:

```

#include <Arduino.h>

void init_temp_hum_task(void);

// For a connection via I2C using the Arduino Wire include:
#include <Wire.h>           // Only needed for Arduino 1.6.5 and earlier
#include "SSD1306Wire.h"    // legacy: #include "SSD1306.h"
// OR #include "SH1106Wire.h" // legacy: #include "SH1106.h"

// For a connection via I2C using brzo_i2c (must be installed) include:
// #include <brzo_i2c.h>      // Only needed for Arduino 1.6.5 and earlier
// #include "SSD1306Brzo.h"
// OR #include "SH1106Brzo.h"

// For a connection via SPI include:
// #include <SPI.h>           // Only needed for Arduino 1.6.5 and earlier
// #include "SSD1306Spi.h"
// OR #include "SH1106Spi.h"

// Optionally include custom images
#include "images.h"

#include "SparkFunHTU21D.h"

//Create an instance of the object
HTU21D myHumidity;

// Initialize the OLED display using Arduino Wire:
SSD1306Wire display(0x3c, SDA, SCL); // ADDRESS, SDA, SCL - SDA and SCL usually populate automatically based on your board's pin
// SSD1306Wire display(0x3c, D3, D5); // ADDRESS, SDA, SCL - If not, they can be specified manually.
// SSD1306Wire display(0x3c, SDA, SCL, GEOMETRY_128_32); // ADDRESS, SDA, SCL, OLEDDISPLAY_GEOMETRY - Extra param required for 12
// SH1106Wire display(0x3c, SDA, SCL); // ADDRESS, SDA, SCL

// Initialize the OLED display using brzo_i2c:
// SSD1306Brzo display(0x3c, D3, D5); // ADDRESS, SDA, SCL
// or
// SH1106Brzo display(0x3c, D3, D5); // ADDRESS, SDA, SCL

// Initialize the OLED display using SPI:
// D5 -> CLK
// D7 -> MOSI (DOUT)
// D0 -> RES
// D2 -> DC
// D8 -> CS
// SSD1306Spi display(D0, D2, D8); // RES, DC, CS
// or
// SH1106Spi display(D0, D2); // RES, DC

#define DEMO_DURATION 3000
typedef void (*Demo)(void);

int demoMode = 0;
int counter = 1;

```

Una vez hecho esto, podemos empezar a darle forma, comenzamos por el setup, el cual se va a encargar de hacer unos prints y inicializar el display para poder usarlo posteriormente.

```

void setup() {
  Serial.begin(115200);
  Serial.println();
  Serial.println();

  Serial.println("HTU21D Example!");

  myHumidity.begin();
//}

//init_temp_hum_task();
// Initialising the UI will init the display too.
display.init();

display.flipScreenVertically();
display.setFont(ArialMT_Plain_10);

}

```

Seguidamente vienen voids que definen varias cosas dentro del display, entre ellas las coordenadas, y muchas otras que no se usan pero se podrían como la implementación de imágenes, etc... Pero siempre hay que tener en cuenta que display usamos ya que el nuestro por ejemplo es monocromático de manera que no se pueden implementar colores en este, solo blanco y negro.

```

void drawFontFaceDemo() {
  // Font Demo1
  // create more fonts at http://oleddisplay.squix.ch/
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  display.setFont(ArialMT_Plain_10);
  display.drawString(0, 0, "Hello world");
  display.setFont(ArialMT_Plain_16);
  display.drawString(0, 10, "Hello world");
  display.setFont(ArialMT_Plain_24);
  display.drawString(0, 26, "Hello world");
}

void drawTextFlowDemo() {
  display.setFont(ArialMT_Plain_10);
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  display.drawStringMaxWidth(0, 0, 128,
    "Lorem ipsum\n dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut l
  )
}

void drawTextAlignmentDemo() {
  // Text alignment demo
  display.setFont(ArialMT_Plain_10);

  // The coordinates define the left starting point of the text
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  display.drawString(0, 10, "Left aligned (0,10)");

  // The coordinates define the center of the text
  display.setTextAlignment(TEXT_ALIGN_CENTER);
  display.drawString(64, 22, "Center aligned (64,22)");

  // The coordinates define the right end of the text
  display.setTextAlignment(TEXT_ALIGN_RIGHT);
  display.drawString(128, 33, "Right aligned (128,33)");
}

void drawRectDemo() {
  // Draw a pixel at given position
  for (int i = 0; i < 10; i++) {
    display.setPixel(i, i);
    display.setPixel(10 - i, i);
  }
}

```

```

display.drawRect(12, 12, 20, 20);

// Fill the rectangle
display.fillRect(14, 14, 17, 17);

// Draw a line horizontally
display.drawHorizontalLine(0, 40, 20);

// Draw a line horizontally
display.drawVerticalLine(40, 0, 20);
}

void drawCircleDemo() {
  for (int i = 1; i < 8; i++) {
    display.setColor(WHITE);
    display.drawCircle(32, 32, i * 3);
    if (i % 2 == 0) {
      display.setColor(BLACK);
    }
    display.fillCircle(96, 32, 32 - i * 3);
  }
}

void drawProgressBarDemo() {
  int progress = (counter / 5) % 100;
  // draw the progress bar
  display.drawProgressBar(0, 32, 120, 10, progress);

  // draw the percentage as String
  display.setTextAlignment(TEXT_ALIGN_CENTER);
  display.drawString(64, 15, String(progress) + "%");
}

void drawImageDemo() {
  // see http://blog.squix.org/2015/05/esp8266-nodemcu-how-to-create-xbm.html
  // on how to create xbm files
  display.drawXbm(34, 14, WiFi_Logo_width, WiFi_Logo_height, WiFi_Logo_bits);
}

Demo demos[] = {drawFontFaceDemo, drawTextFlowDemo, drawTextAlignmentDemo, drawRectDemo, drawCircleDemo, drawProgressBarDemo, drawImageDemo};
int demoLength = (sizeof(demos) / sizeof(Demo));
long timeSinceLastModeSwitch = 0;

```

Finalmente el loop, el cual se va a encargar de mostrar por el display y pantalla todo lo necesario, temperatura, humedad, el texto, etc...

```

void loop() {

    float humd = myHumidity.readHumidity();
    float temp = myHumidity.readTemperature();

    Serial.print("Time:");
    Serial.print(millis());
    Serial.print(" Temperature:");
    Serial.print(temp, 1);
    Serial.print("C");
    Serial.print(" Humidity:");
    Serial.print(humd, 1);
    Serial.print("%");

    Serial.println();

    // clear the display
    display.clear();
    // draw the current demo method
    //demos[demoMode]();

    display.setTextAlignment(TEXT_ALIGN_CENTER);
    display.setFont(ArialMT_Plain_10);
    display.drawString(128/2, 0, "HUMEDAD");
    display.setFont(ArialMT_Plain_16);
    display.drawString(128/2, 11, String(humd)+ "%");
    display.setFont(ArialMT_Plain_10);
    display.drawString(128/2, 30, "TEMPERATURA");
    display.setFont(ArialMT_Plain_16);
    display.drawString(128/2, 41, String(temp)+ "°C");

    display.setFont(ArialMT_Plain_10);
    display.setTextAlignment(TEXT_ALIGN_RIGHT);
    display.drawString(128, 54, String(millis()/3600000)+String(":")\
        +String((millis()/60000)%60)+String(":")\
        +String((millis()/1000)%60));

    // write the buffer to the display
    display.display();

    delay(100);
}

```