

## TRABAJO #1

### DISEÑO ELECTRÓNICO

Se realizó la integración de un sensor de temperatura y humedad DHT11 y una ESP32 TTGO OLED v1 para enviar datos a un servidor alojado en Ubidots. El código y las evidencias de funcionamiento se presentan a continuación:

```
#include <UbiConstants.h>
#include <UbiTypes.h>
#include <UbidotsEsp32Mqtt.h>
#include "DHT.h"
#include <TFT_eSPI.h>
#include <SPI.h>

/*****
 * Se definen las constantes
 *****/
#define DHTPIN 12      // Pin donde está conectado el sensor DHT11
#define DHTTYPE DHT11 // Tipo de sensor utilizado (DHT11)
const char *UBIDOTS_TOKEN = "BBUS-UmLWlxEkCI6nYJel8pIL3zd96IPJa"; // Token de Ubidots
const char *WIFI_SSID = "Lasso"; // Nombre de la red Wi-Fi
const char *WIFI_PASS = "24825207"; // Contraseña de la red Wi-Fi
const char *DEVICE_LABEL = "esp32"; // Etiqueta del dispositivo en Ubidots
const char *TEMP_LABEL = "temp"; // Etiqueta para la variable de temperatura
const char *HUMIDITY_LABEL = "humidity"; // Etiqueta para la variable de humedad

const int PUBLISH_FREQUENCY = 5000; // Frecuencia de envío en milisegundos

unsigned long timer;

// Se inicializa el sensor DHT
DHT dht(DHTPIN, DHTTYPE);

// Se inicializa el cliente de Ubidots
Ubidots ubidots(UBIDOTS_TOKEN);

// Se inicializa la pantalla OLED
TFT_eSPI tft = TFT_eSPI(135, 240); // Pantalla de la TTGO OLED

/*****
 * Se definen las funciones auxiliares
 *****/
void callback(char *topic, byte *payload, unsigned int length)
```

```

{
  Serial.print("Mensaje recibido [");
  Serial.print(topic);
  Serial.print("]: ");
  for (int i = 0; i < length; i++)
  {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

void initDisplay()
{
  // Se inicializa la pantalla y se configura su estado inicial
  tft.init();
  tft.setRotation(1);
  tft.fillScreen(TFT_BLACK);
  tft.setTextSize(2);
  tft.setTextColor(TFT_GREEN, TFT_BLACK);
  tft.setTextDatum(MC_DATUM);
  tft.drawString("Inicializando...", tft.width() / 2, tft.height() / 2);
  delay(2000);
  tft.fillScreen(TFT_BLACK);
}

/*****
* Se definen las funciones principales
*****/

void setup()
{
  // Se configura el puerto serial para depuración
  Serial.begin(115200);

  // Se inicia el sensor DHT
  dht.begin();

  // Se inicializa la pantalla OLED
  initDisplay();

  // Se establece la conexión con la red Wi-Fi y Ubidots
  ubidots.connectToWifi(WIFI_SSID, WIFI_PASS);
  ubidots.setCallback(callback);
  ubidots.setup();
}

```

```

ubidots.reconnect();

timer = millis();
}

void displayData(float temperature, float humidity)
{
    // Se actualizan los datos en la pantalla OLED
    tft.fillScreen(TFT_BLACK);
    tft.setTextSize(2);
    tft.setCursor(0, 20);
    tft.setTextColor(TFT_GREEN, TFT_BLACK);

    tft.printf("Temp: %.2f C\n", temperature);
    tft.printf("Hum: %.2f %%\n", humidity);

    Serial.printf("Temp: %.2f C, Hum: %.2f %%\n", temperature, humidity);
}

void loop()
{
    // Se verifica la conexión a Ubidots y se reconecta si es necesario
    if (!ubidots.connected())
    {
        ubidots.reconnect();
    }

    // Se envían datos periódicamente
    if (abs((long)(millis() - timer)) > PUBLISH_FREQUENCY)
    {
        float temperature = dht.readTemperature(); // Se lee la temperatura
        float humidity = dht.readHumidity(); // Se lee la humedad

        // Se verifica si las lecturas son válidas
        if (isnan(temperature) || isnan(humidity))
        {
            Serial.println("Error al leer el sensor DHT11");
            tft.fillScreen(TFT_RED);
            tft.setTextDatum(MC_DATUM);
            tft.drawString("Error sensor!", tft.width() / 2, tft.height() / 2);
        }
        else
        {
            // Se muestran los datos en la pantalla OLED

```

```
displayData(temperature, humidity);
```

```
// Se envían los datos a Ubidots
```

```
ubidots.add(TEMP_LABEL, temperature); // Se agrega la temperatura
```

```
ubidots.add(HUMIDITY_LABEL, humidity); // Se agrega la humedad
```

```
ubidots.publish(DEVICE_LABEL); // Se publican los datos
```

```
}
```

```
timer = millis(); // Se reinicia el temporizador
```

```
}
```

```
ubidots.loop();
```

```
}
```

