

Micropython WeatherStation

Project Documentation

12/05/2024



Author: Manuel Portero Leiva

introducción

El objetivo de este documento es mostrar las funcionalidades del dispositivo creado, exponer las diferentes partes, el montaje y el código utilizado para el montaje de nuestra estación meteorológica. Este documento separará el montaje electrónico, el código de configuración de la tarjeta Arduino y el diseño 3d de la carcasa utilizada.

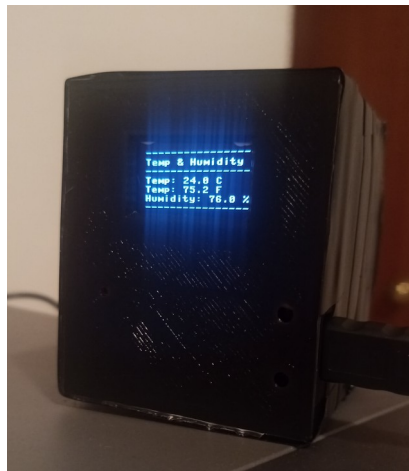


Figura 1: Estación meteorológica Arduino

Montaje electrónico

Los componentes del dispositivo, así como su montaje, se muestra a continuación.

Componentes:

- ESP8266/ESP32 x1
- Cable USB para conectar el ESP con ordenador.
- Sensor DHT11 x1
- Dispositivo pantalla OLED x 1
- Cables tipo "Jumper" para conectar las entradas de sensores y periféricos a la placa Arduino.

Conexiones del circuito:

La placa, ESP8266/ESP32 Procesa la información del sensor para mostrarla en el display. El sensor DHT11 leerá la temperatura y humedad del ambiente. Las conexiones del dispositivo DHT11 se muestra a continuación:

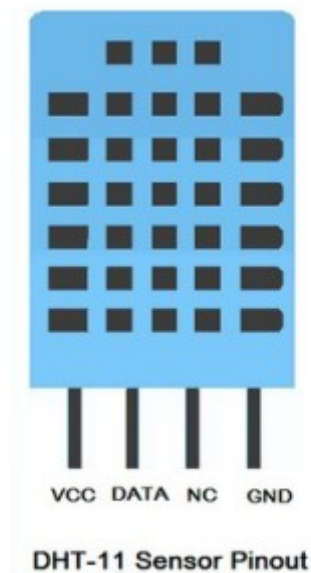


Figura 2: Conexiones sensor DHT11

Para realizar las conexiones del DHT11 con ESP8266, se debe de conectar VCC y GND pins del DHT11 con 3V y GND del ESP8266. Conectar el pin de datos del DHT11 con cualquier GPIO of ESP8266 (en este proyecto al GPIO14 (D5) del ESP8266).

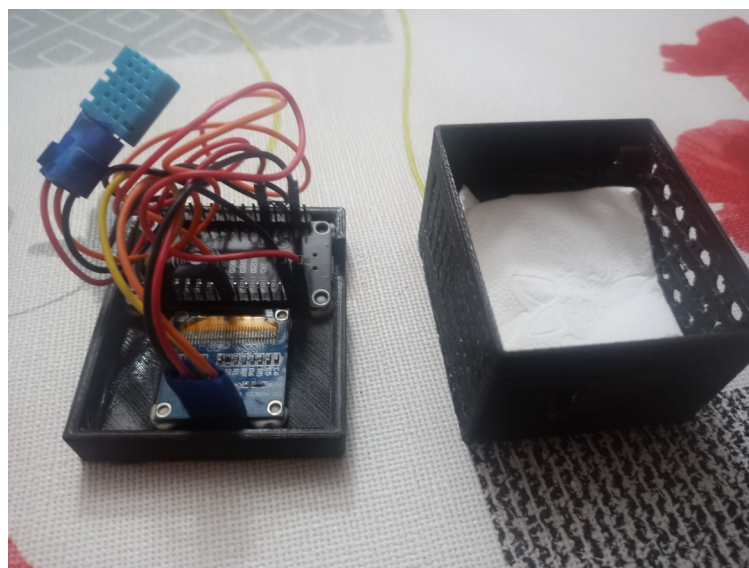


Figura 3: Conexiones sensor dispositivo

Para conectar la pantalla OLED el pin SCL y SDA iran respectivamente al pin 5 y 4 de la placa y la conexión de voltaje se realizara a cualquier punto de voltaje 3,3v y a toma de tierra

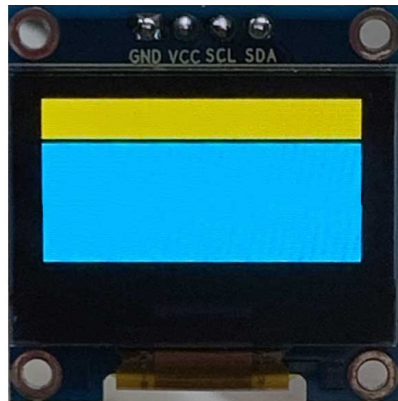


Figura 4: Conexiones dispositivo pantalla

Código placa Esp8622

El código micropython utilizado en este proyecto se muestra a continuación:

```
from machine import Pin, SoftI2C
from time import sleep
import dht
import network
import urequests as requests
import ujson
import ssd1306
```

```
sensor = dht.DHT11(Pin(14))
```

```
while True:
    try:
        sleep(2)
        sensor.measure()
```

```

temp = sensor.temperature()
hum = sensor.humidity()
temp_f = temp * (9/5) + 32.0
print('Temperature: %3.1f C' %temp)
print('Temperature: %3.1f F' %temp_f)
print('Humidity: %3.1f %%' %hum)

i2c = SoftI2C(scl=Pin(5), sda=Pin(4))

oled_width = 128
oled_height = 64
oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)

oled.text('-----', 0, 0)
oled.text('Temp & Humidity', 0, 10)
oled.text('-----', 0, 20)
oled.text('Temp: %3.1f C' %temp, 0, 30)
oled.text('Temp: %3.1f F' %temp_f, 0, 40)
oled.text('Humidity: %3.1f %%' %hum, 0, 50)
oled.text('-----', 0, 60)
oled.show()

except OSError as e:
    print('Failed read or send information.')

```

Diseño 3d de carcasa

Las carcasas utilizadas para realizar en este proyecto son de licencia creative commons y se pueden encontrar en el siguiente repositorio de código:

[Link diseños](#)

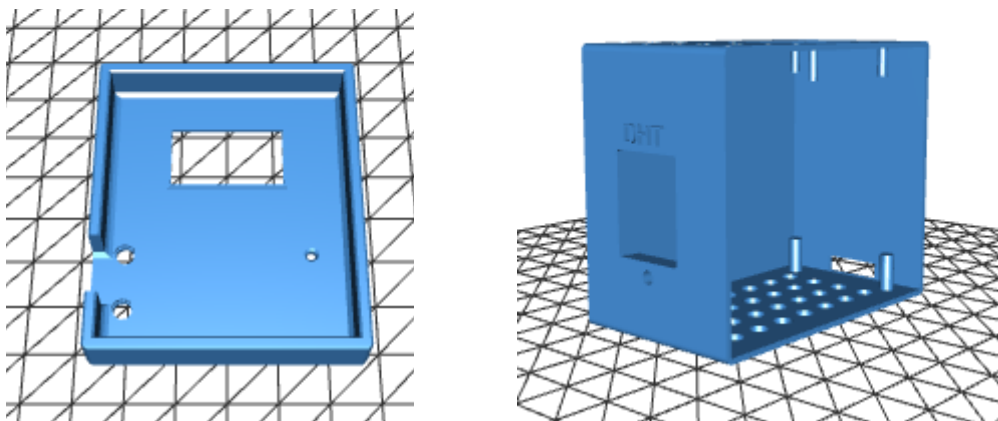


Figura 4: Imágenes diseño de carcasas

Referencias

- <https://www.engineersgarage.com/micropython-esp8266-esp32-online-weather-station/>
- <https://randomnerdtutorials.com/micropython-oled-display-esp32-esp8266/>