

KATOLÍCKA UNIVERZITA V RUŽOMBERKU
PEDAGOGICKÁ FAKULTA

**ESP32 – Meranie teploty a tlaku zobrazené na OLED displeji
a internetovej stránke**

Záverečný projekt

2022/2023

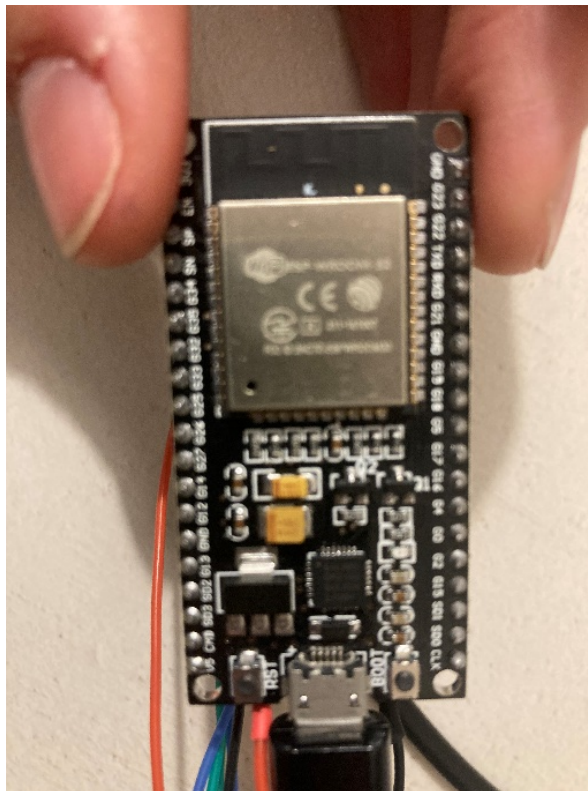
Mgr. Ján Hamadej (2RŠI)

Cieľom tohto projektu bolo pomocou ESP32 a senzora (čidlá) teploty a vlhkosti zostrojiť zariadenie, ktoré nám pomocou OLED displeja a WiFi siete bude monitorovať teplotu a vlhkosť vzduchu. Tieto hodnoty v reálnom čase nám zobrazí na OLED displeji a vytvorenej internetovej stránke.

Použité pomôcky:

- ESP 32 s WiFi
- USB kábel
- Senzor vlhkosti a teploty (SHT 30)
- OLED displej (SSD1306)

Obrázky použitých pomôcok:

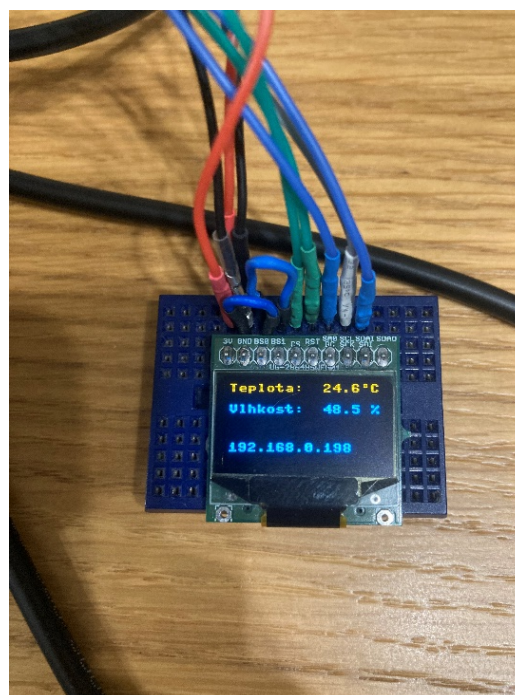


Obr. 1: ESP 32 s WiFi



Obr. 2: Senzor vlhkosti a teploty (SHT 30)

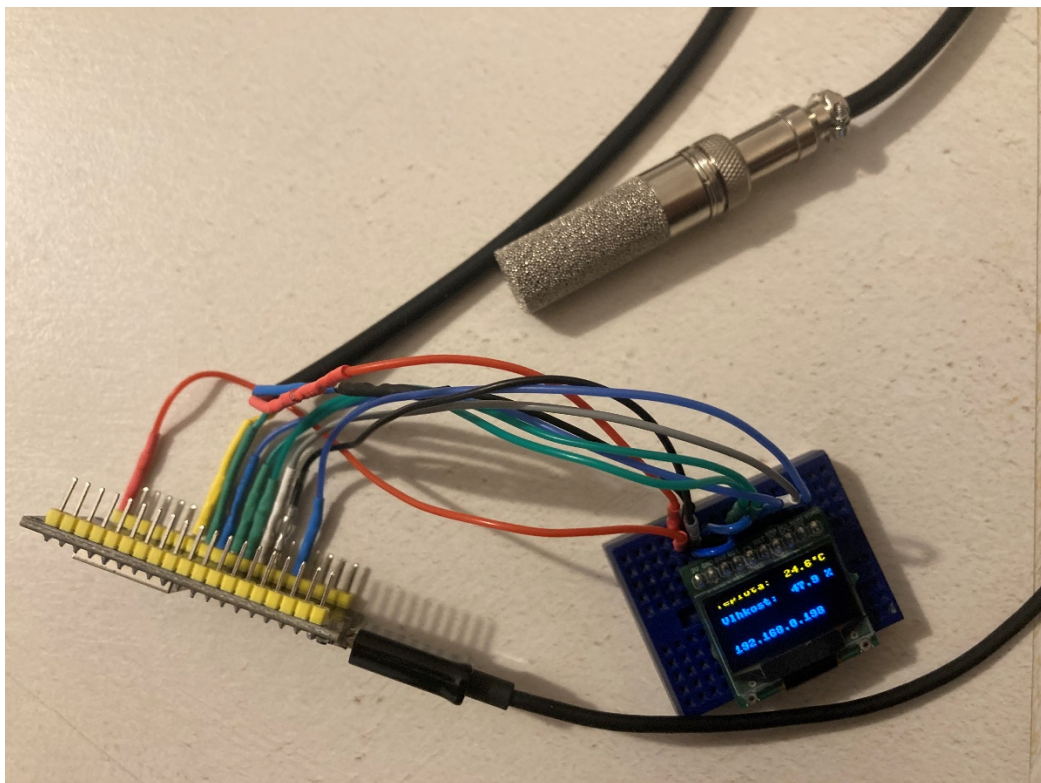
- senzor SHT30 - cez zbernicu I2C:
 - SCL (nožička ESP32 - GPIO32)
 - SDA (nožička ESP32 - GPIO33)
- a napájanie: GND a +3V



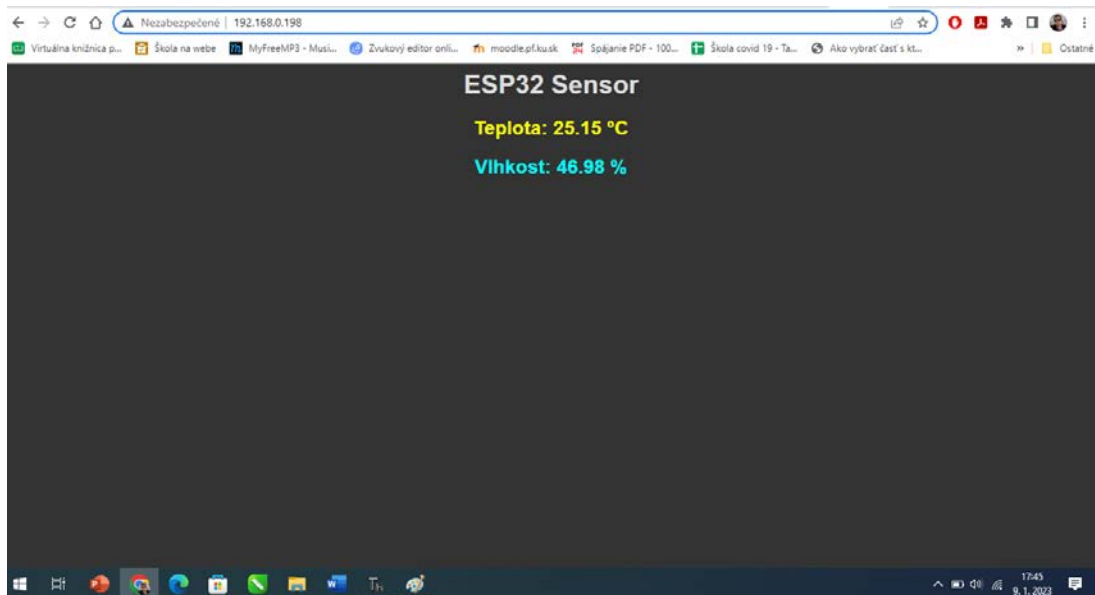
Obr.3: OLED displej (SSD1306)

- displej
- cez zbernicu SPI + ďalšie riadiace signály:
 - SCK: ESP32 - GPIO14
 - SDA: ESP32 - GPIO13
 - DC: ESP32 - GPIO25
 - CS: ESP32 - GPIO27
 - RST: ESP32 - GPIO26
- displej je technológie OLED, ma rozlíšenie 128x64 pixelov, je žltá - modrá (prvých 16 pixelových riadkov je žltých a zvyšok modrý)

Dokumentácia prepojenia komponentov a funkčnosti internetovej stránky



Obr. 4: Prepojenie ESP32 so senzorom vlhkosti, teploty a OLED displejom



Obr. 5: Internetová stránka 192.168.0.198

Zdrojový kód

boot.py:

```
import network

wlan = network.WLAN(network.STA_IF)

def do_connect():
    wlan.active(True)
    if not wlan.isconnected():
        print('connecting to network...')
        wlan.connect('Nazov WIFI', 'Heslo Wifi')
        while not wlan.isconnected():
            pass
    print('NETWORK:', wlan.ifconfig())

do_connect()
```

main.py:

```
import machine
```

```
import ssd1306
```

```
import sht30
```

```
import uweb
```

```
cs = machine.Pin(27)
```

```
rst = machine.Pin(26)
```

```
dc = machine.Pin(25)
```

```
sck = machine.Pin(14)
```

```
mosi = machine.Pin(13)
```

```
sda = machine.Pin(33, machine.Pin.IN, machine.Pin.PULL_UP)
```

```
scl = machine.Pin(32, machine.Pin.IN, machine.Pin.PULL_UP)
```

```
spi = machine.SPI(1)
```

```
i2c = machine.I2C(0, scl=scl, sda=sda, freq=10000)
```

```
oled = ssd1306.SSD1306_SPI(128, 64, spi, dc, rst, cs)
```

```
def read_sensor():
```

```
    try:
```

```
        temp_c, temp_f, humidity = sht30.read(i2c)
```

```
    except OSError as e:
```

```
        print('read_sensor:', e)
```

```
        return None
```

```
    print(f'{temp_c:6.2f}°C, {temp_f:6.2f}°F, {humidity:6.2f}%')
```

```
    oled.fill(0)
```

```
oled.text(f'Teplota:{temp_c:6.1f} C', 0, 0, 1)
oled.text(f'Vlhkost:{humidity:6.1f} %', 0, 16, 1)
oled.text(wlan.ifconfig()[0], 0, 48, 1)
oled.rect(115, 0, 3, 3, 1)
oled.show()
return temp_c, humidity
```

```
HTML = ""<html>
```

```
<head>
```

```
<title>ESP32 Sensor</title>
```

```
<meta http-equiv="Content-type" content="text/html; charset=utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<meta http-equiv="refresh" content="10">
```

```
<link rel="icon" href="data:,">
```

```
<style>
```

```
html {{
```

```
font-family: Helvetica;
```

```
display:inline-block;
```

```
margin: 0px auto;
```

```
text-align: center;
```

```
background-color: #333;}}
```

```
h1 {{color: #ddd;}}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>ESP32 Sensor</h1>
```

```
<p>
```

```

        <h2 style="color: #ff0;">Teplota: {temperature:6.2f} °C</h2>
        <h2 style="color: #0ff;">Vlhkost: {humidity:6.2f} %</h2>
    </p>
</body>
</html>"""

```

```

def send_response(client, temperature, humidity):
    html = HTML.format(temperature=temperature, humidity=humidity)
    uweb.response(client, data=html)

```

```

def main():
    sock = uweb.web_server()

    temperature = 0
    humidity = 0

    while True:
        web_res = uweb.web_wait(sock)
        sensor_res = read_sensor()
        if sensor_res:
            temperature, humidity = sensor_res
        if web_res:
            client, addr, method, url, headers = web_res
            print(f'{addr[0]:s}:{addr[1]:d} {method:s} {url:s}')
            send_response(client, temperature, humidity)

```



```
if __name__ == '__main__':  
    main()
```

Použité zdroje:

<https://docs.micropython.org>

<https://www.w3schools.com/>

<https://uniot.sk/>