



Dokumentace k projektu  
**Meteostanice**  
Mikroprocesorové a vestavěné systémy - IMP

19. prosince 2025

Jakub Lůčný (xlucnyj00)

# **Obsah**

<b>1</b>	<b>Úvod</b>	<b>2</b>
1.1	Použité technologie . . . . .	2
<b>2</b>	<b>Návrh</b>	<b>2</b>
2.1	Zapojení . . . . .	2
2.2	Hlavní tok programu . . . . .	2
<b>3</b>	<b>Implementační detaily</b>	<b>4</b>
3.1	Struktura programu . . . . .	4
<b>4</b>	<b>Závěr</b>	<b>5</b>
<b>5</b>	<b>Použitá literatura</b>	<b>6</b>

# 1 Úvod

Dokumentace popisuje návrh a implementaci meteostanice pro zobrazení informací o teplotě a vlhkosti vzduchu s využitím vývojové desky ESP32, senzoru SHT31 a OLED displeje. Program také umožňuje volitelně se připojit na WiFi síť a odesílat naměřené hodnoty pomocí protokolu MQTT.

## 1.1 Použité technologie

**Hardware:**

- Vývojová deska **ESP32**
- Senzor teploty a vlhkosti **SHT31** - pro měření hodnot
- **OLED displej** - pro zobrazení naměřených hodnot

**Software:**

- **ESP-IDF** framework s **PlatformIO** rozšířením pro VSCode
- **u8g2** knihovna pro ovládání displeje
- **HiveMQ Public Broker** - veřejně dostupný MQTT broker, na který zařízení posílá naměřená data
- **MQTT Explorer** - pro zobrazení dat, která byla odeslána na HiveMQ Broker

# 2 Návrh

Tato sekce popisuje celkový návrh řešení problému od zapojení jednotlivých periferií až po hlavní tok programu.

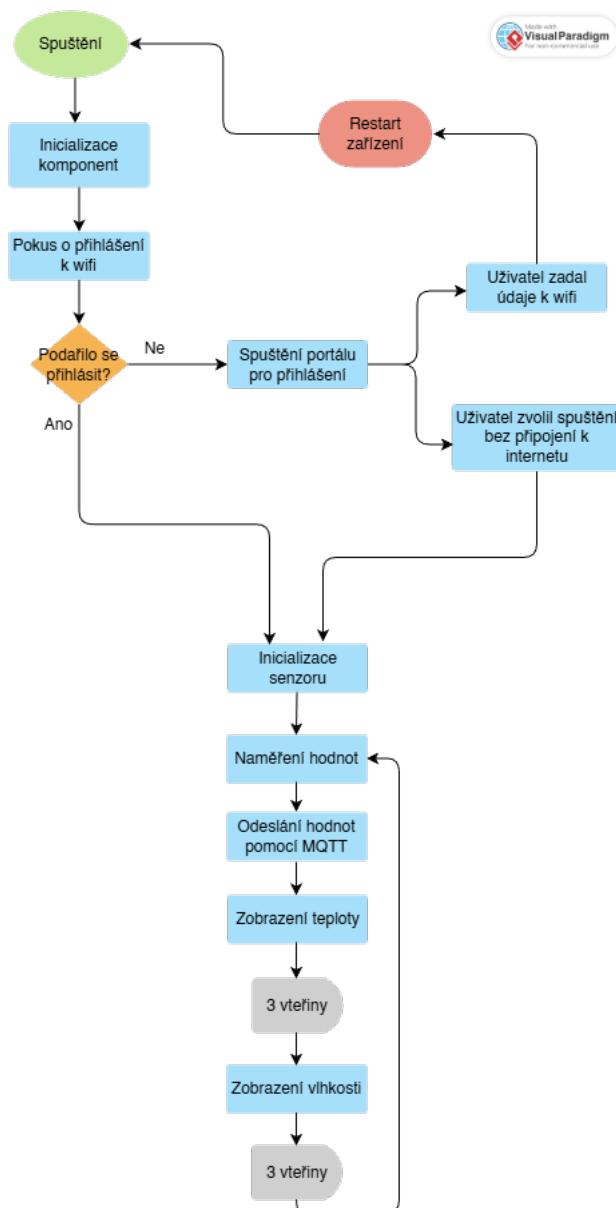
## 2.1 Zapojení

Komunikace s oběma periferiemi probíhá přes I2C sběrnici. Tedy obě periferie jsou připojeny ke zdroji napětí 3.3V a uzemnění na vývojové desce a dále obě periferie sdílí připojení jejich datového SDA pinu k SDA pinu 21 desky a také připojení SCL pinu pro zdroj hodin této komunikace a to k pinu 22 desky.

## 2.2 Hlavní tok programu

1. **Spuštění:** při spuštění se jako první inicializují jednotlivé komponenty jako NVS úložiště (Non-volatile storage), I2C sběrnice, displej a WiFi.
2. **WiFi:** hned poté se pokusí zařízení přihlásit na WiFi síť, jejíž údaje má uložené v NVS.
  - 2.1. **Konfigurace WiFi:** pokud se nepodaří přihlásit nebo zařízení nemá uložené žádné předešlé údaje, spustí "WiFi přístupový bod"(AP) a portál pro zadání údajů a zobrazí informace o portálu na displeji.

- 2.2. **Portál:** pro přístup ke konfiguračnímu portálu se musí uživatel připojit k přístupovému bodu desky a spustit prohlížeč s adresou uvedenou na displeji. Poté má možnost buď zadat SSID a heslo od WiFi nebo vybrat možnost spuštění bez připojení k internetu.
3. **Měření:** poté už se inicializuje samotný senzor a spustí se nekonečná smyčka programu.
  4. **Nekonečná smyčka:** ve smyčce se přibližně každých 6 sekund opakuje - získání naměřených údajů, odeslání dat na MQTT broker (pokud je zařízení připojeno k internetu), zobrazení teploty na 3 sekundy a nakonec zobrazení vlhkosti vzduchu na další 3 sekundy. Zobrazované hodnoty jsou doplněny o ikonky a *progressbar*.



Obrázek 1: Flow diagram programu

### 3 Implementační detaily

Implementace je realizována za pomocí ESP-IDF frameworku a z velké části se opírá o veřejně dostupné příklady[1], odkud byla převzata a mírně poupravena značná část projektu, a to převážně co se inicializace a správného nastavení jednotlivých komponent týká.

Implementace je rozdělena do několika souborů. Většina vlastního autorského kódu se nachází v souboru `main.c`, ostatní soubory obsahují převážně převzaté a mírně upravené kusy kódů z již zmíněných veřejných příkladů[1].

Pro ovládání displeje byla použita knihovna **u8g2**, která umožňuje snadné zobrazování jak textu, tak i například grafických ikonek apod.

Údaje jsou na displeji zobrazovány společně s ikonkou a *progressbarem*. Ikonky byly převzaty jako volně dostupné a pomocí nástroje[5] převedeny do bitmapové reprezentace, kterou následně zobrazí knihovna pro ovládání displeje. *Progressbar* zobrazuje čas do přepnutí na další údaj.

Převedení hodnot získaných ze senzoru na odpovídající údaje o teplotě a vlhkosti bylo podle následujících vzorců[4]:

$$T \text{ [}^{\circ}\text{C}] = -45 + 175 \cdot \frac{S_T}{2^{16} - 1}$$

$$RH = 100 \cdot \frac{S_{RH}}{2^{16} - 1}$$

Data jsou odesílána pomocí MQTT na adresu brokeru `mqtt://broker.hivemq.com:1883` jako téma (*topic*) `meteostanice/measurements`. Tyto přednastavené hodnoty se dají změnit definováním příslušných maker nebo změněním hodnoty těchto maker na začátku souboru `main.c`. Data jsou odesílána ve formátu JSON. Příklad formátu dat:

```
{  
    "temp_c": 22.18,  
    "hum": 67.8  
}
```

#### 3.1 Struktura programu

Program se skládá z následujících částí:

- `display.c`, `display.h` - inicializace a obsluha displeje, funkce pro vykreslování
- `i2c_bus.c`, `i2c_bus.h` - inicializace I2C sběrnice
- `main.c` - hlavní modul, řízení programu
- `mqtt.c`, `mqtt.h` - inicializace mqtt klienta a publikování hodnot
- `portal.c`, `portal.h` - nastavení a ovládání konfiguračního portálu

- `sensor.c`, `sensor.h` - inicializace senzoru a čtení hodnot
- `wifi.c`, `wifi.h` - inicializace wifi a nvs modulu, načtení přihlašovacích údajů

## 4 Závěr

Meteostanici se podařilo implementovat podle návrhu a nejsou známy žádné chyby či nedostatky. Řešení by bylo snadné případně rozšířit o snímání a zobrazování i dalších hodnot. Díky exportu dat je možná další vizualizace či jiné zpracování naměřených hodnot. Implementace je dostatečně robustní a i uživatelsky přívětivá díky informacím o aktuálním stavu zobrazovaným na displeji v průběhu konfigurace připojení.

Program demonstруje jak základní snímání hodnot z digitálního senzoru, tak jednoduchou i pokročilejší vizualizaci grafiky na displeji, ale i použití WiFi přístupového bodu, WiFi připojení a MQTT protokolu pro přenos dat. Mimo to projekt také zahrnoval základní propojení I2C periferií a správné nastavení I2C sběrnice pro komunikaci.

## 5 Použitá literatura

- [1] Espressif Systems. ESP-IDF Examples. Available: <https://github.com/espressif/esp-idf/blob/master/examples>
- [2] Last Minute Engineers. ESP32 OLED Display with Arduino IDE. Available: <https://lastminuteengineers.com/oled-display-esp32-tutorial/>
- [3] HiveMQ. Introducing the MQTT Protocol – MQTT Essentials: Part 1. Available: <https://www.hivemq.com/blog/how-to-get-started-with-mqtt/>
- [4] Sensirion. Datasheet SHT3x. Available: [https://www.laskakit.cz/user/related\\_files/sht3x.pdf](https://www.laskakit.cz/user/related_files/sht3x.pdf)
- [5] javl. image2cpp. Available: <https://javl.github.io/image2cpp/>