

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Mikroprocesorové a vestavěné systémy
Chytrá domácnost

Veronika Novikova(xnovik03)

1. Cíl projektu

Cílem tohoto projektu bylo navrhnout a implementovat ovládací panel pro chytrou domácnost pomocí **ESP32** a **SSD1306 OLED displeje**. Panel umožňuje uživatelům:

- **Číst aktuální stav** prvků chytré domácnosti, jako je osvětlení a teplota v místnostech.
- **Nastavovat hodnoty** prvků, například zapnutí/vypnutí světla nebo změnu teploty.
- **Interagovat přes tlačítka**.

Komunikace mezi zařízením a externím serverem je realizována pomocí **MQTT protokolu**, který umožňuje obousměrný přenos dat mezi serverem a ESP32. Celé řešení bylo vyvinuto v prostředí **PlatformIO** v rámci **Visual Studio Code**, s využitím **ESP-IDF** frameworku.

2. Použité technologie a nástroje

1. **ESP32** vývojová deska
2. **SSD1306 OLED** displej
3. **MQTT** protokol.
4. **PlatformIO** a **Visual Studio Code**
5. **Mosquitto**

3. Prezentace projektu

https://vutbr-my.sharepoint.com/:v:/g/personal/xnovik03_vutbr_cz/EaYycv5JIRpLkVXjj_eekuIBK9ATbZdF-BLSFH7OVmnCIA?e=dYeKCj

4. Hardwarové zapojení

4.1 Komponenty

- **ESP32:** Hlavní řídící jednotka.
- **SSD1306 OLED displej (SPI):** Používá se pro vizualizaci dat a interakci s uživatelem.
- **Tlačítka SELECT a OK:** Slouží pro navigaci(levé) a potvrzování v menu(pravé).
- **Napájení:** ESP32 je napájeno přes USB.

4.2 Pinové zapojení

MOSI (Data)	GPIO 23
SCLK (Hodiny)	GPIO 18
CS (Chip Select)	GPIO 5
DC (Data/Command)	GPIO 27
RESET	GPIO 17
Tlačítko SELECT	GPIO 34
Tlačítko OK	GPIO 35

5. Struktura aplikace

5.1 Přehled systému

Aplikace je rozdělena do několika stavů, které určují aktuální obrazovku a možnosti ovládání:

- **Hlavní menu:** Výběr mezi zobrazením stavu a provedením změn.

- **Zobrazení stavu:** Přehled všech místností, včetně teploty a stavu osvětlení.
- **Nastavení místnosti:** Úprava hodnot jednotlivých prvků místnosti (světlo, teplota).
- **Úprava osvětlení:** Přepnutí stavu světla (zapnuto/vypnuto).
- **Úprava teploty:** Zvýšení teploty.

5.2 Základní funkce

5.2.1 Inicializace

1. **init_rooms()**
Inicializuje výchozí hodnoty pro všechny místnosti (teplota na minimum, světla vypnuta).
Popis: Připravuje data pro každou místnost.
2. **init_buttons()**
Konfiguruje GPIO piny pro tlačítka SELECT a OK jako vstupy.
Popis: Zajišťuje správné fungování tlačítek.

5.2.2 Zpracování vstupů

3. **handle_buttons()**
Sleduje stav tlačítek a provádí akce podle aktuálního stavu systému.
 - SELECT: Posun v menu nebo změna hodnoty.
 - OK: Potvrzení volby nebo návrat do předchozí úrovně.**Popis:** Umožňuje uživatelskou interakci s menu a funkcemi.

5.2.3 Zobrazení dat

4. **update_display()**
Dynamicky aktualizuje obsah na OLED displeji na základě aktuálního stavu aplikace.
Popis: Zobrazuje menu, stav místností nebo možnosti nastavení.

5.2.4 Komunikace s MQTT

5. **send_room_state_to_server()**
Publikuje aktuální stav místnosti na MQTT server.
Popis: Odesílá JSON zprávy s informacemi o osvětlení a teplotě.
6. **fetch_room_state_from_server()**
Přihlašuje ESP32 k odběru MQTT témat, která odpovídají jednotlivým místnostem.
Popis: Zajišťuje synchronizaci dat mezi serverem a ESP32.

7. `handle_mqtt_data()`

Zpracovává příchozí MQTT zprávy a aktualizuje stav místností.

Popis: Implementace logiky pro příjem a zpracování dat.

6. Funkce systému

6.1 Uživatelské rozhraní

- **Menu:**
 - Hlavní obrazovka nabízí volbu mezi „Zobrazit stav“ a „Proveď změny“.
 - Uživatel se naviguje tlačítkem SELECT a potvrzuje tlačítkem OK.
- **Displej:**
 - Přehledné zobrazení všech místností (teplota, stav světla).
 - Možnost nastavení hodnot jednotlivých prvků.

6.2 Komunikace

- **MQTT protokol:**
 - Publikace stavu místností na témata ve tvaru `home/room/{číslo_místnosti}`.
 - Příjem dat pro aktualizaci stavu místností.

7. Testování komunikace s MQTT serverem

Nástroj: `mosquitto_sub` (MQTT klient pro příjem zpráv).

Postup: Spustili jsme příkaz pro odběr všech zpráv z témat :

```
mosquitto_sub -h mqtt.eclipseprojects.io -t "home/room/#" -v
```

Ověřila jsem příchozí zprávy z ESP32 na serveru:

Zprávy obsahovaly aktuální stav světla a teploty v místnostech.

Příklad záznamu:

```
home/room/1 {"light":1,"temperature":18}  
home/room/2 {"light":1,"temperature":17}  
home/room/4 {"light":0,"temperature":16}  
home/room/4 {"light":0,"temperature":17}  
home/room/4 {"light":1,"temperature":21}
```

1. Popis výsledků:

- Místnost 1: Světlo je zapnuto, teplota 18 °C.
 - Místnost 4: Světlo bylo postupně vypnuto, teplota se zvýšila na 21 °C a poté se světlo zapnulo.
2. Ověřila jsem, že MQTT zprávy jsou **publikovány jako retained**, což umožnilo, aby po restartu ESP32 byly k dispozici poslední známé hodnoty.

8. Popis souborů

1. **main.c**

- Obsahuje hlavní implementaci aplikace, včetně logiky displeje, obsluhy tlačítek a MQTT komunikace.

2. **my_data.h**

- Uchovává přihlašovací údaje pro Wi-Fi (SSID a heslo) pro snadnou změnu bez zásahu do hlavního kódu.

3. **components**

- Knihovna pro ovládání displeje SSD1306 přes SPI, stažená z [nopnop2002 GitHub repozitáře](#).

4. **config/sdkconfig**

- Generovaný konfigurační soubor pro ESP-IDF, obsahující nastavení projektu (GPIO piny a Wi-Fi parametry).

5. **CMakeLists.txt**

- Konfigurační soubor pro sestavení projektu pomocí CMake v ESP-IDF.

9. Využité zdroje pro projekt

Při tvorbě tohoto projektu byly využity následující zdroje:

1. **Oficiální dokumentace ESP-IDF**

- [MQTT dokumentace](#): Informace o použití MQTT klienta v ESP-IDF.
- [HTTP Server](#): Dokumentace k implementaci a konfiguraci HTTP serveru na ESP32.

2. **Mosquitto MQTT**

- [Mosquitto dokumentace](#): Použito pro testování MQTT serveru, odběr zpráv a kontrolu správnosti publikovaných dat.

3. **GitHub**

- [nopnop2002 GitHub repozitář](#): Knihovna pro ovládání SSD1306 OLED displeje a základní práci s fonty.

4. YouTube

- [ESP-IDF Programming with MQTT](#)
- [ESP-IDF Basics: SSD1306 OLED Display](#)