



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Měření intenzity světla a řízení svitu LED diody s odesíláním dat do MQTT

Projekt z předmětu:
Mikroprocesorové a vestavěné systémy

Autor: Daniil Zverev (xzvere00)
Datum: 15. prosince 2024

Úvodní popis projektu

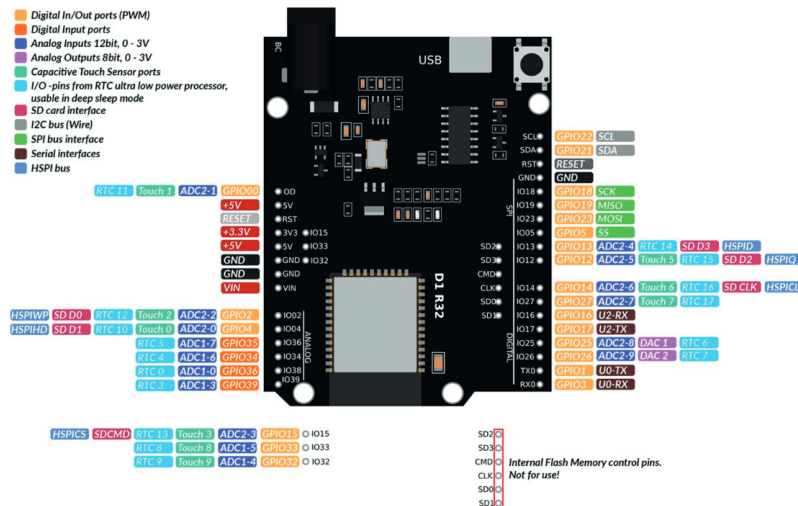
Cílem projektu je vytvořit aplikaci pro platformu ESP32, která:

- Měří úroveň okolního světla (v luxech) pomocí světelného senzoru (např. BH1750).
- Na základě naměřených hodnot světla lineárně upravuje jas LED diody, aby při nedostatku světla LED dioda svítila jasněji a při dostatečném okolním osvětlení byla utlumená či zcela vypnutá.
- Umožňuje uživatelskou konfiguraci klíčových parametrů (práh, hystereze, MQTT topic) pomocí příkazů zadávaných přes sériovou konzoli (UART).
- Ukládá nastavení do NVS (Non-Volatile Storage), aby zůstalo zachováno i po restartu.
- Pravidelně odesílá aktuální naměřenou hodnotu světla na MQTT broker, kde lze hodnoty sledovat z jiného zařízení či aplikace.

Použitý hardware

Projekt byl vyvíjen a testován na desce typu **D1 R32 Board**, což je deska založená na ESP32 s výstupy a piny kompatibilními s Arduino prostředím. Disponuje integrovanou Wi-Fi, podporou I2C, PWM a dalšími periferiemi. Její pinout je znázorněn na obrázku níže:

D1 R32 Board Pinout



Popis způsobu řešení

Celý projekt je rozdělen do více zdrojových souborů pro přehlednost a modularitu. Níže je uveden podrobný popis odpovědnosti jednotlivých souborů:

Hlavní soubor projektu:

main.c

- Inicializace NVS, Wi-Fi, LED, senzor, MQTT a UART.
- Načtení uložených konfigurací (práh, hystereze, MQTT topic) z NVS.
- Spuštění **sensor_task**, který periodicky:
 - Čte hodnoty ze senzoru světla.
 - Na základě prahu a naměřené hodnoty vypočítá jas LED.
 - Nastaví jas LED diody.
 - Publikuje aktuální hodnotu luxu jako JSON na MQTT broker.

Konfigurace a ukládání do NVS:

config.c/h

- Poskytuje funkce pro načtení a uložení konfigurace do NVS.
- `load_settings_from_nvs()` načítá prah, hysteresi a MQTT topic. Pokud neexistují, nastaví výchozí hodnoty.
- `save_settings_to_nvs()` ukládá aktuální konfiguraci do NVS.

LED a její řízení jasu:

`led.c/h`

- Inicializace PWM (LEDC) pro ovládání jasu LED.
- `led_init()` nastaví PWM timer a kanál.
- `led_set_brightness(float level)` nastaví jas LED (0.0 - 1.0) včetně gamma korekce pro lineární vnímání.

Čtení hodnoty z senzoru:

`sensor.c/h`

- Ovládání a čtení hodnot z BH1750 senzoru přes I2C.
- `sensor_init()` inicializuje I2C a zapne kontinuální měření.
- `sensor_read_lux()` čte hodnotu v luxech z BH1750.

Wi-Fi připojení:

`wifi.c/h`

- Zajišťuje připojení ESP32 k Wi-Fi síti.
- Po získání IP adresy inicializuje MQTT klienta.

MQTT komunikace:

`mqtt.c/h`

- Inicializuje MQTT klienta a připojí se k brokeru.
- `mqtt_init(const char* uri)` nastaví a spustí MQTT klienta.
- `mqtt_publish(const char* topic, const char* msg)` publikuje zprávu na daný topic.

UART příkazy:

`uart_cmd.c/h`

- Umožňuje nastavit parametry přes UART příkazy:
 - `threshold <hodnota>`
 - `hysteresis <hodnota>`
 - `mqtt_topic <řetězec>`
 - `show` - zobrazí aktuální konfiguraci
- Po změně uložení nové konfigurace do NVS.

Závěrečné shrnutí

Projekt splňuje požadavky:

- Měří úroveň osvětlení a upravuje jas LED diody podle nastaveného prahu s hysteresí.
- Umožňuje uživatelskou úpravu parametrů přes UART.
- Uchovává nastavení v NVS pro přetrvání po restartu.
- Publikuje naměřené hodnoty do MQTT pro vzdálený přístup.

Celé řešení je modulární a snadno rozšiřitelné.

Demonstrace práce

Pro ukázkou praktického fungování projektu byla připravena krátká videoprezentace, která ukazuje:

- Připojení k Wi-Fi a následné odesílání hodnot do MQTT.
- Změnu nastavení přes UART a okamžitou reakci na nově zadané hodnoty.
- Úpravu jasu LED na základě okolního osvětlení.

Odkaz na demonstrační video : https://youtube.com/shorts/YmhAsS1_CSo?feature=share