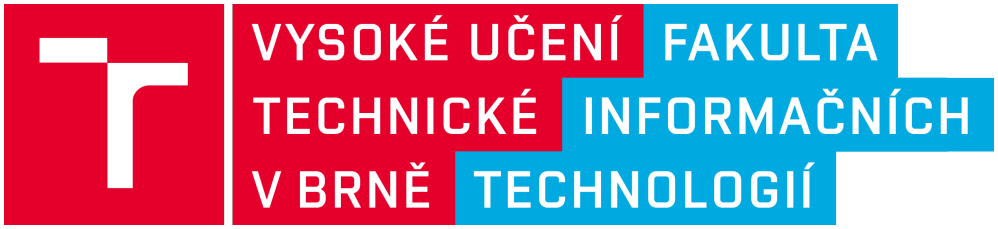

Dokumentace k projektu z předmětu IMP

Téma: Spínání světla dle intenzity

13. prosince 2023

Kambulat Alakaev xalaka00

Obsah

1 Úvod a příprava

Projekt spočíval v sestavení programu pro mikrokontroler ESP-32 s použitím senzoru intenzity světla (BH1750) a LED diody. Pro přenos dat senzor používá rozhraní I2C, což je synchronní sériové rozhraní s protokolem master-slave, které komunikuje způsobem half-duplex.

S rostoucí intenzitou světla padající na senzor by se měl snižovat jas diody. Úroveň jasu světla lze řídit pomocí PWM (pulzně šířková modulace).

PWM je technika, která se používá v elektronice k řízení výkonu dodávaného elektrickému zařízení nebo komponentě. Tato technika pracuje s pulsy elektrického signálu, a to zejména s modifikací délky trvání pulzů.

Základní myšlenkou PWM je rychlý přepínající signál, kde pulzy mají proměnlivou šířku. Doba trvání pulzu, známá jako "šířka pulzu,"se mění tak, aby efektivně ovlivnila průměrný výkon dodávaný zařízení. Tím lze dosáhnout řízení jasu, otáček motoru, teploty a dalších parametrů elektrických zařízení.

Projekt byl napsán v jazyce C s využitím frameworku ESPIDF(Espressif) a vývojového prostředí PlatformIO jako rozšíření pro Visual Studio Code. Pro inspiraci byly použity open-source zdroje[2], dokumentace k ESPIDF(Espressif)[1] a PlatformIO [4] a obecné informace k projektu [3].

2 Realizace

Aby všechny periferní prvky správně fungovaly, musely být na začátku nakonfigurovány.

Pro LED diodu bylo nutné nakonfigurovat PWM časovač a jeho parametry: frekvence, rozlišení pracovního cyklu(duty cycle) signálu PWM, režim rychlosti(speed mode) pro práci na velké/malé frekvence.

Dále bylo nutné připravit a nastavit konfiguraci kanálu LEDC PWM pro generování vlnového signálu. To zahrnovalo nastavení GPIO pinu pro výstup, výběr kanálu a časovače. Dále bylo třeba provést inicializaci a konfiguraci sběrnice I2C: nastavení modu na "master"pro řízení periferie(senzoru), provést nastavení pinů a frekvence. Pak bylo nutné senzor zapnout a nastavit do provozu. K tomu bylo nutné provést následující kroky:

- Vytvořit I2C příkazový handler, který se bude používat k vytvoření a odeslání I2C příkazů senzoru.
- Vygenerovat I2C start podmínku, která označuje začátek komunikace na I2C sběrnici.
- Odeslat byte s adresou zařízení na I2C sběrnici pro zahájení komunikace se senzorem BH1750 v režimu zápisu.
- Poslat na sběrnici I2C hodnotu 0x10. Tato hodnota je příkazem pro senzor BH1750, aby začal měřit intenzitu světla.
- Pak je třeba dokončit komunikaci a odeslat všechny příkazy ve frontě na sběrnici I2C v režimu "master".

Pomocí těchto kroků byl senzor nastaven tak, aby zachycoval hodnotu intenzity světla. Nyní je třeba přečíst hodnotu intenzity ze snímače. Za tímto účelem byla vytvořena tzv. úloha(task), která běží v nekonečné smyčce a snímá hodnotu intenzity ze senzoru.

Kroky potřebné ke čtení hodnoty intenzity jsou v některých místech podobné těm, které byly popsány výše, ale režim zápisu by měl být změněn na režim čtení, dále by měly být načteny 2 bajty (horní a dolní) a zkombinovat je, abychom získali intenzitu v *luxech*(fotometrická jednotka intenzity osvětlení). Pak podle dokumentaci k senzoru BH1750 [5] získanou hodnotu bylo nutno vydělit 1.2, což je pravděpodobně korekční nebo škálovací faktor, který se aplikuje na data získaná ze senzoru BH1750 za účelem jejich převodu na přesnější hodnotu v luxech. Tímto postupem můžeme ze snímače odečíst hodnotu intenzity světla.

Pak podle zadání máme linearizovat získanou hodnotu osvětlení. Hodnota v luxech byla namapována na hodnotu odpovídající poměru doby, po kterou LED dioda svítí během celého periodického signálu a které se říká *duty*.

Pak bylo nutné umožnit uživateli zadat do terminálu hodnotu odpovídající horní prahové hodnotě intenzity světla, při které se má dioda vypnout (přestat svítit). K tomuto účelu bylo použito rozhraní UART (Universal

Asynchronous Receiver-Transmitter), které představuje možnost provádět asynchronní komunikaci. Nejprve bylo nutné nakonfigurovat UART: nastavit přenosovou rychlost(baud rate), počet bitů pro data, paritu a počet stop bitů. Po nastavení a konfiguraci UART je možné číst data ze vstupu terminálu. Získaná prahová hodnota zadaná uživatelem bude použita později.

Podle zadání by měl mikrokontroler udržovat prahovou hodnotu zaslanou uživatelem i po vypnutí zařízení. K tomu potřebujeme použít modul NVS (Non Volatile Storage), který nám umožňuje ukládat data do mikrokontroleru. Abychom mohli zapsat hodnotu do NVS máme ho nejprve inicializovat a pak otevřít pro zápis a po zapsání hodnoty předané uživatelem je nutné zápis potvrdit a NVS modul uzavřít. Při každém restartu mikrokontroleru se z paměti NVS načte uložená hodnota horní prahové intenzity světla.

Podle zadání musí být při změně hodnoty poměru *duty* zohledněna tzv. hystereze. Hystereze se projevuje tím, že výstupní hodnoty nejsou jednoznačně závislé pouze na skutečném vstupu, ale také na tom, jakým způsobem této hodnoty dosáhly. Příkladem může být zapnutí nebo vypnutí zařízení na základě určité prahové hodnoty. Když vstupní hodnota překročí práh a zařízení se zapne, lze hysterezi nastavit tak, že se zařízení vypne až tehdy, když vstupní hodnota klesne pod jiný, nižší práh, než při kterém bylo spuštěno.

Tato funkcionalita byla implementována tak, že se jas diody změní pouze tehdy, je-li skutečná hodnota intenzity světla větší než součet uživatelem definované prahové hodnoty a hodnoty hystereze, nebo je-li skutečná hodnota intenzity světla menší než rozdíl mezi uživatelem definovanou prahovou hodnotou a hodnotou hystereze.

Zadání vyžadovalo odeslání aktuální hodnoty intenzity do zprostředkovatele MQTT, který tuto hodnotu zobrazí.K tomu bylo nutné nastavit připojení mikrokontroleru k internetu (WiFi). Vzhledem k omezením pro připojení zařízení k internetu na kolejích VUT bylo rozhodnuto připojit zařízení k telefonu, který poskytoval mobilní internet a fungoval jako router (přístupový bod na internet). Pro připojení k MQTT brokeru a nastavení komunikace bylo nutné nastavit URI a port.

3 Odkaz na video

Omlouváme se za vodoznak ve videu, ale je průhledný a vše je dobře vidět. V dolní části videa jsou komentáře k prováděným krokům.

Odkaz: https://youtu.be/SzEwXLfxNKc

Odkazy

[1] Ltd. Espressif Systems (Shanghai) Co. *ESP-IDF Programming Guide*. URL: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/.

[2] *espressif/esp-idf*. URL: https://github.com/espressif/esp-idf/tree/master/examples/.

[3] Ph.D. Ing. Vojtěch Mrázek. *Projekt - ESP32 (M)*. URL: https://moodle.vut.cz/mod/page/view.php?id=338880.

[4] PlatformIO. *PlatformIO IDE for VSCode*. URL: https://docs.platformio.org/en/latest/integration/ide/vscode.html.

[5] ROHM Semiconductor. *Digital 16bit Serial Output Type Ambient Light Sensor IC*. URL: https://www.laskakit.cz/user/related_files/bh1750fvi-e-186247.pdf.