

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Inteligentné osvetlenie pracovného stola

Semestrálny projekt

Autor: Bc. Miroslav Hájek

Študijný program: Inteligentné softvérkové systémy

Predmet: Vnorené systémy

Akademický rok: 2022 / 2023

ZADANIE SEMESTRÁLNEHO PROJEKTU

Predmet: **VNORENÉ SYSTÉMY**

Riešiteľ: **Bc. Miroslav Hájek**

Školský rok: **2022/2023**

Názov projektu: **Inteligentné osvetlenie pracovného stola**

Zadanie:

Navrhnite a zrealujte systém inteligentného svietidla pre pracovný stôl, ktoré sa bude prispôsobovať okolitému prostrediu a zapínať sa pri prítomnosti človeka v blízkosti. Systém bude postavený na báze mikrokontroléra ESP32, ktorý bude ovládať intenzitu a farbu RGB stmavovateľného LED pásu. Svietidlo sa bude prispôsobovať dennému svetlu, tým že po zotmení sa samočinne aktivuje v prípade pohybu v miestnosti. Mobilná aplikácia pre Android bude schopná cez Bluetooth nastaviť stav zopnutia svetla, jeho farebnú teplotu, jas a prahovú úroveň vonkajšieho osvetlenia.

Projekt musí obsahovať:

1. Zadanie
2. Analýzu problematiky
3. Návrh riešenia
4. Opis implementácie
5. Otestovanie riešenia
6. Zhodnotenie výsledkov
7. Zoznam použitej literatúry
8. Technickú dokumentáciu

Termín odovzdania: 4.5.2023

V Bratislave, 2.3.2023

doc. Ing. Tibor Krajčovič, PhD.
prednášateľ

Obsah

1	Analýza problematiky	1
1.1	Špecifikácia požiadaviek	1
1.2	Riadiaca jednotka	1
1.3	Senzory a akčné členy	3
1.3.1	RGB LED pásik	3
1.3.2	PIR pohybový senzor	4
1.3.3	Senzor okolitého osvetlenia	4
2	Návrh riešenia	5
2.1	Diagramy zapojenia	5
2.2	Mobilná aplikácia	6
2.3	Spínanie svetla	7
3	Opis implementácie	8
4	Otestovanie riešenia	11
4.1	Test: Opakovane nadväzovanie Bluetooth spojenia	12
4.2	Test: Aktívna detekcia pohybu	12
4.3	Test: Deaktivovaná detekcia pohybu	13
4.4	Test: Manuálne zapnutie a vypnutie svetla	13
4.5	Test: Zmena farebnej teploty svetla	14
4.6	Test: Zmena jasu svetla	14
4.7	Test: Reakcia svetla na okolité osvetlenie	15
4.8	Test: Načítanie pôvodných nastavení po zapnutí	15
5	Zhodnotenie výsledkov	16
6	Technická dokumentácia	18
6.1	Prehľad hardvérových súčiastok	18
6.2	Zdrojové súbory pre firmvér	19
6.3	Zdrojové súbory pre Android aplikáciu	20
6.4	Syntax Bluetooth príkazov	21

1 Analýza problematiky

Zariadenie na ovládanie svietidla sa má skladať z dvoch nezávislých integrovaných častí: riadiaca jednotka s firmvérom priamo regulujúca spínanie lampy podľa senzorov, a mobilná aplikácia na zmenu aktívnych nastavení svetla a detekcie pohybu komunikujúca cez Bluetooth. V sekcií popíšeme výber vhodných súčiastok a základné princípy ich fungovania.

1.1 Špecifikácia požiadaviek

Zo zadania vyplýva niekoľko funkcionálnych požiadaviek na systém:

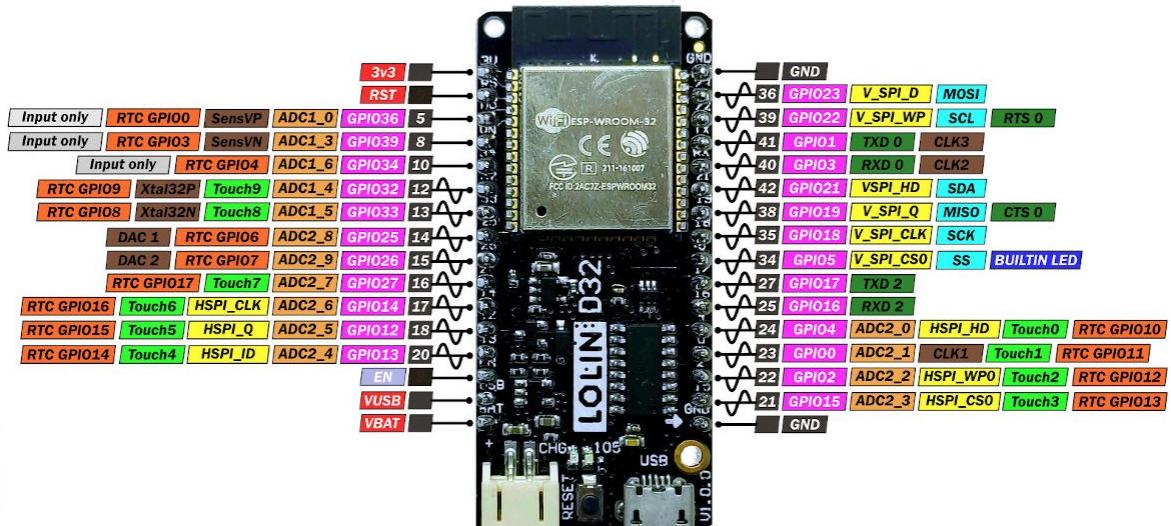
- Svietidlo rovnomerne osvetlí celý povrch pracovného stola.
- Svietidlo sa zapne pri slabom osvetlení a súčasne detekcii pohybu v miestnosti.
- Používateľ je schopný cez mobilnú aplikáciu vypnúť/zapnúť svetlo, nastaviť farebnú teplotu, jas a prahovú úroveň pre slabé osvetlenie.
- Používateľské nastavenia svietidla sa zachovajú medzi zapnutiami zariadenia.
- Vnorený systém kontroléra svetla je postavený na mikrokontroléri ESP32.
- Mobilná aplikácia je určená na platformu Android a komunikuje cez Bluetooth.

1.2 Riadiaca jednotka

Mikrokontrolér **ESP32 WeMos Lolin D32** (Obr. 1) sme zvolili spomedzi ostatných bežne dostupných vývojových dosiek (ako sú Arduino, Raspberry Pi, STM32, BeagleBone a pod.) hlavne pre podporu pre viacerých štandardných mikroprocesorových rozhraní na rozličných pinoch. Doska je postavená na module *ESP32-WROOM-32* [1], ktorý disponuje procesorom s frekvenciou 80 MHz (až do 240 MHz) a pamäťou 4 MB Flash a 520 kB SRAM príliš nelimitujúce rozhodnutia na účely prototypu.

Napájacie napätie dosky je 3,3 V, ale cez USB konektor môžeme pripojiť zdroj 5 V. ESP32 sa vyznačuje aj dobrou podporou od výrobcu v podobe SDK obsahujúce ovládače na všetky dostupné rozhrania pre periférie ESP-IDF [2], ktorý tiež zahŕňa operačný systém reálneho času FreeRTOS.

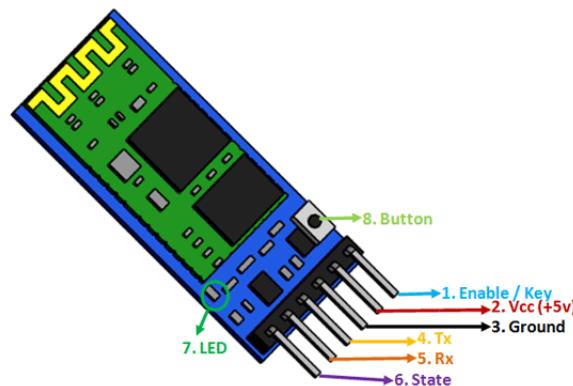
ESP32 sice disponuje konektivitou na WiFi a Bluetooth, ale funkcie pre RFCOMM nie sú zatiaľ implementované, preto sa Bluetooth komunikácia musí realizovať exter-



Obr. 1: Vývojová doska ESP32 WeMos Lolin D32 [3]

ným **modulom HC-05** (Obr. 2). Posielanie a príjem správ sa uskutočňuje cez *UART* protokol, kde piny RXD a TXD musia byť na napäťových úrovniach 3,3 V, zatiaľ čo napájanie je v rozmedzí 3,6 - 6 V (zväčša 5 V). Baudová rýchlosť je predvolene určená na 9600.

Pin Enable slúži na prepnutie modulu do AT príkazového režimu (high úroveň), v ktorom je možné upraviť napr. baud. Pin State signalizuje stav Bluetooth pripojenia a je zároveň zapojený na indikačnú LED, ktorá sa môže nachádzať v 3 stavoch: (i) bliknutie raz za 2 sekundy - príkazový režim, (ii) blikanie 2-krát za 1 sekundu - pripojenie nadviazané, (iii) opakované blikanie - čakanie na pripojenie [4].



Obr. 2: Bluetooth modul HC-05 [4]

1.3 Senzory a akčné členy

Regulovateľné LED diódy v úlohe akčného členu musia reagovať na situáciu v okolitom prostredí, menovite detekciu pohybu človeka v miestnosti (cez infračervené čidlo) a intenzitu vonkajšieho osvetlenia (cez fotodiódou).

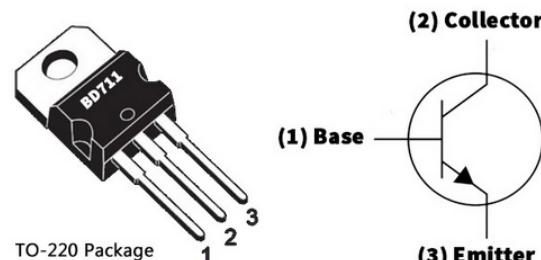
1.3.1 RGB LED pásik

Typ LED pásiku sa vyberá podľa rôznych kritérii ako sú účel použitia, či ide o dekoráciu alebo má slúžiť na osvetlenie miestnosti. Na základe toho prispôsobíme svetelný výkon, farbu, vodovzdornosť (stupeň ochrany krytom), rozmery, z čoho nakoniec vyplýva cieľové napätie najčastejšie 12 V, 24 V, 230 V.

Rozlišujeme pásiky podľa technológie produkcie teplotného odtieňu: jednofarebné, CCT - s dvoma typmi diód 2700 K a 6500 K a reguláciou medzi nimi, RGB - s štvorpišnové LED diódami. Svietivosť pásiku ovplyvňuje príkon udávaný na meter, na nasvietenie napr. kuchynskej linky sa odporúča LED pásik s vyšším príkonom 12 - 20 W/m, tie je žiaduce namontovať na hliníkový podklad, z dôvodu chladenia. Ďalší faktor určujúci charakter svetla je rozloženie diód a ich počet na meter pásika [5].



(a) LED pásik



(b) NPN tranzistor BD711

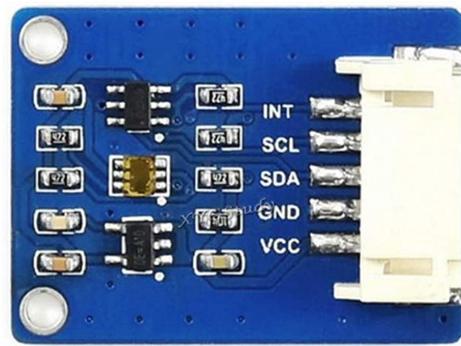
Obr. 3: Akčné členy svietidla

Vhodný kandidát na interiérové osvetlenie stola sa ukazuje **RGB LED pásik** (Obr. 3a) so samolepiacou fóliou 3M 300LSE bez krytia IP20, napäťom zdroja 12 V, príkonom 14,4 W/m a hustotou diód 60 ks/m, ktorý vyprodukuje až 550 lm/m svetla [6]. LED diódy na pásiku sú zapojenie tri do série so spoločnou anódou v 10 cm oddeľiteľných úsekokach. Vetva pre červenú farbu má navyše zapojený do série 300 Ω SMD rezistor, ostatné vetvy majú zapojené 150 Ω rezistor.

Jednotlivé farebné kanály nemôžeme spínať a regulovať priamo s pinu mikroprocesoru, pretože nie je prítomné rovnaké napätie: 3,3 V voči 12 V, a nepostačuje ani maximálny prúd $I_{OL} = 28$ mA. Dedikovaný tranzistor bude spínať prúd do 1,2 A ($I = 14,4W/12V$), preto potrebujeme výkonovejšie tranzistory než z bežne používaneho radu BC ($I_C < 100mA$). **NPN tranzistor BD711** (Obr. 3b) dokáže spínať prúd až do 12 A [7].



(a) PIR senzor pohybu SB312



(b) Senzor osvetlenia TSL25911

Obr. 4: Senzory s označením vývodov

1.3.2 PIR pohybový senzor

Na detekciu pohybu ľudí sa používajú infračervené PIR senzory líšiace sa veľkosťou, citlivosťou a nastaviteľnosťou úrovne detektora. Zlepšenie snímacích vlastností je docieľené plastovou Fresnelovou šošovkou na senzore. **Olimex SB312** (Obr. 4a) sa vyznačuje kompaktnosťou s rozmermi PCB: 10 x 8 mm. Disponuje uhlom snímacieho kužela do 100° s dosahom 3 - 5 metrov. Oneskorenie výstupu voči detegovanému pohybu je 2 sekundy a po rovnako dlhý čas drží výstupnú logickú úroveň v jednotke. Napájacie napätie senzora je 3,3 V [8].

1.3.3 Senzor okolitého osvetlenia

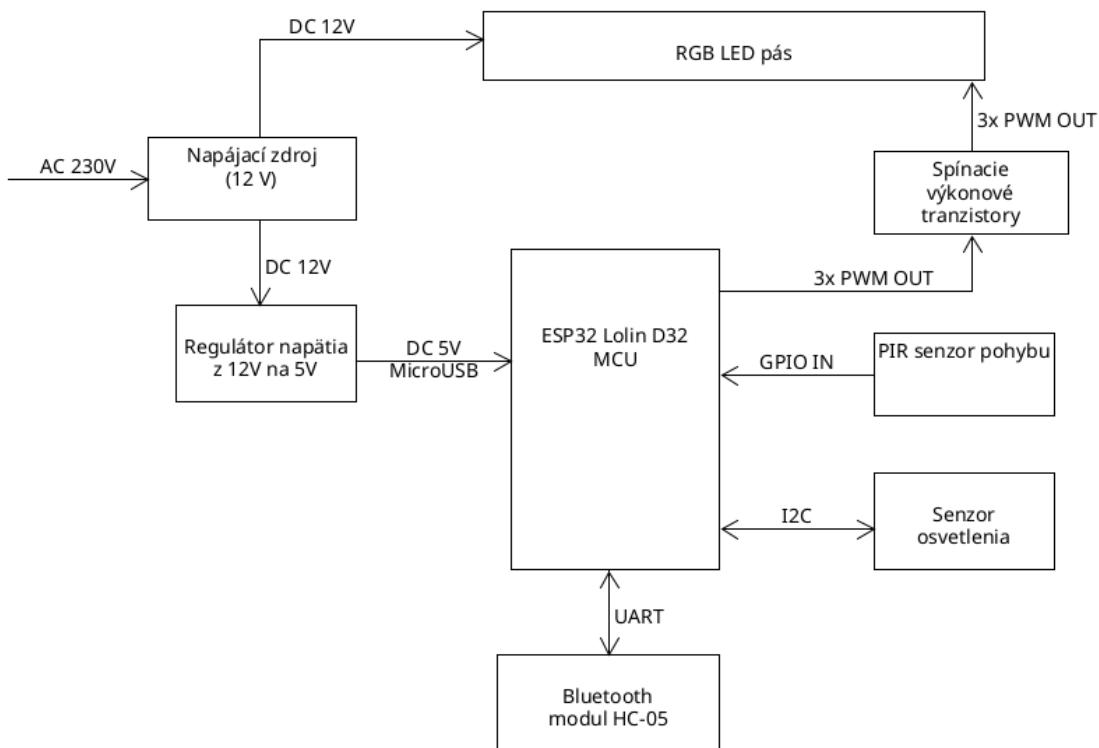
Modul senzora osvetlenia Waveshare WS-17146 (Obr. 4b) obsahuje obvod TSL25911FN, ktorý samostatne prerátava nameranú intenzitu svetla na fotodióde do luxov s použitím vzorca na approximáciu ľudského videnia. Senzor pracuje na napäti 3,3 V aj 5 V a komunikuje cez I2C zbernicu do 400 kbit/s na adresu 0x29. Najdôležitejšie registre pre odčítanie aktuálnej hodnoty sú Control (0x01) a ALS Data (0x14 - 0x17) [9].

2 Návrh riešenia

V časti návrhu uvažujeme ohľadom zapojenia hardvérových súčiastok z pohľadu napäťových úrovní aj riadiacich signálov v systéme. Spôsob čítania zo senzorov a ovládania ovplyvňujú ovládače adaptérov používané firmvérom a ich vzájomnú koordináciu. Voľbu nastavení, ktoré môže používateľ zmeniť je následne potrebné zahrnúť do prototypu obrazovky mobilnej aplikácie.

2.1 Diagramy zapojenia

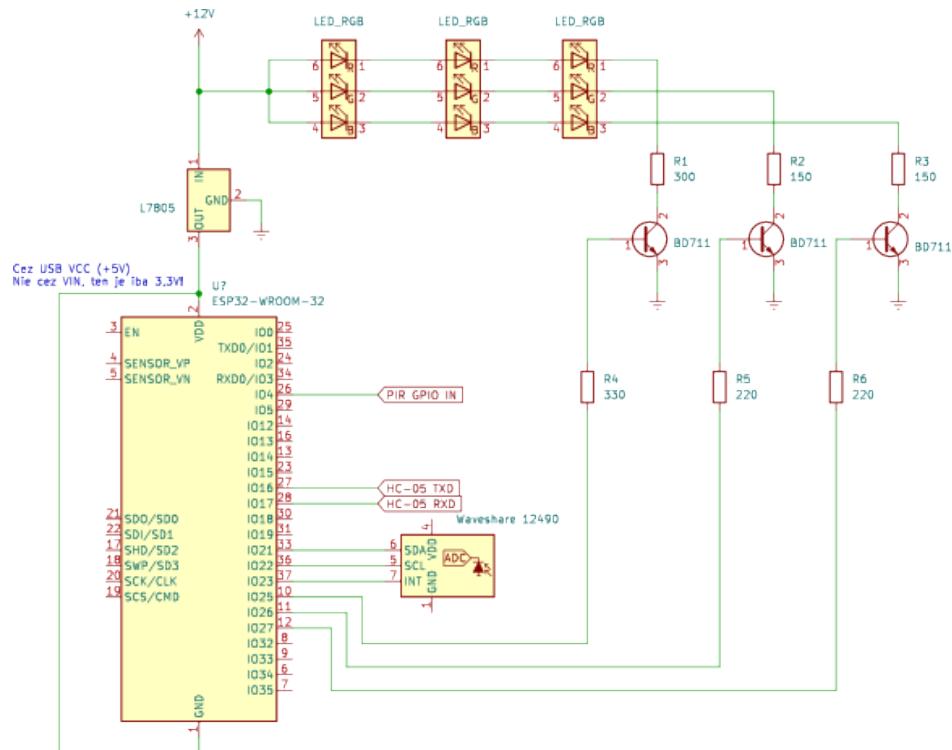
Vnorený systém inteligentnej stolovej lampy vychádza z **blokovej schémy** na Obr. 5. Zariadenie je napájané z elektrickej siete 230 V AC, ktoré napajiací zdroj transformuje na 12 V DC priamo použiteľných pre LED pásiak. Mikrokontrolér potrebuje napätie 3,3 V, ale USB konektor umožňuje pripojenie priamo na 5 V DC získaných z lineárneho regulátora napäťia.



Obr. 5: Blokový diagram vnoreného systému inteligentného svietidla

Periférie komunikujú s ESP32 cez štandardné digitálne zbernice na napäťovej úrovni 3,3 V: UART pre Bluetooth modul, I2C pre senzor osvetlenia. Intenzita LED diód je

regulovaná cez tri samostatné PWM kanály (pulzne-šírková modulácia) a senzor pohybu je pripojený ako vstup na GPIO pin.



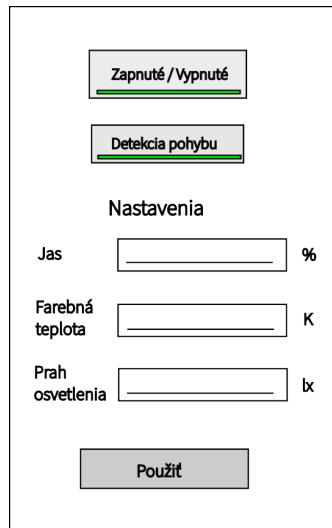
Obr. 6: Schéma obvodu

Na schéme 6 je zakreslené zapojenie jednotlivých modulov na piny mikrokontroléra. Piny VCC a GND zo senzorov sú vedené na piny 3,3 V a GND na vývojovej doske ESP32. Bluetooth modul je pripojený priamo na 5 V z L7805. LED pásič už obsahuje rezistory pripojené na kolektory tranzistorov. Bázové rezistory sú zapojené ako prídavné súčiastky.

2.2 Mobilná aplikácia

Stav a aktuálne vlastnosti lampy sa budú nastavovať cez rozhranie mobilnej aplikácie určenej na platformu Android. Wireframe rozloženia tlačidiel a textových polí na vpisovanie čísel sa nachádza na Obr. 7.

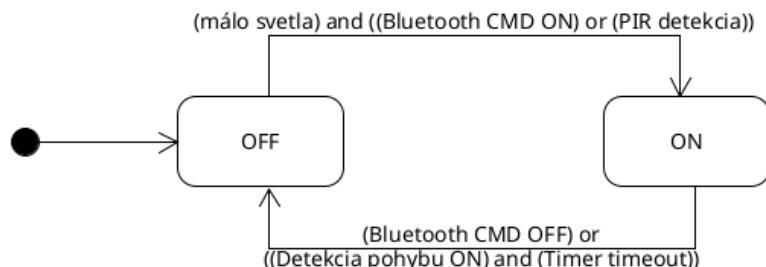
Funkcionality aplikácie budú dostupné až po úspešnom nadviazaní spojenia s modulom HC-05 cez Bluetooth. Pri pripojení na svietidlo si aplikácia načíta nastavenia uložené na mikrokontroléri do textových polí. Tlačidlo „Použit“ umožní odoslať naraz všetky nastavenia z aplikácie. Na tlačidlách zapnutia svetla a detekcie pohybu, alebo vedľa nich, bude indikovaný aktuálny stav oznamovaný zo zariadenia pri akejkoľvek zmene.



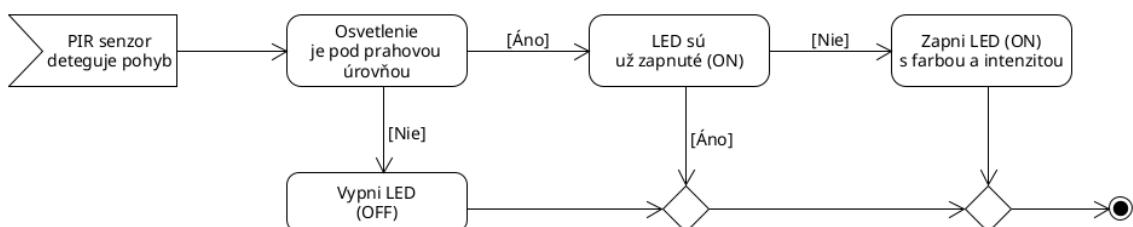
Obr. 7: Wireframe mobilnej aplikácie

2.3 Spínanie svetla

Svetlo sa môže v podstate nachádzať len v dvoch stavoch - zapnuté a vypnuté, aj keď charakter emitovaného svetla môže byť odlišný od situácie. Existuje však viacero scenárov na prepnutie stavu pri návrhu ktorých, je dôležité odstrániť konflikty. Nesmie nastať jav, že sa svetlo rozsvieti aktivovaním jedného pravidla a vzápäť sa zhasne uplatnením iného.



Obr. 8: Stavový diagram spínania svietidla



Obr. 9: Zapnutie svietidla pri detekcii pohybu

Pravidlá na prenutie stavov svetla sú nasledovné (Obr. 8):

- **Ak je OFF:** Okolité osvetlenie musí byť pod nastavenou prahovou úrovňou: $\text{lux} \leq \text{threshold}$, a súčasne bol zaslaný dopyt o zapnutie svetla, buď z mobilnej aplikácie alebo od pohybového senzora.
- **Ak je ON:** Mobilná aplikácia poslala dopyt na vypnutie svetla alebo vypršal časovač od posledného dopytu na zapnutie (ten je aktívny vždy keď je spustená detekcia pohybu).

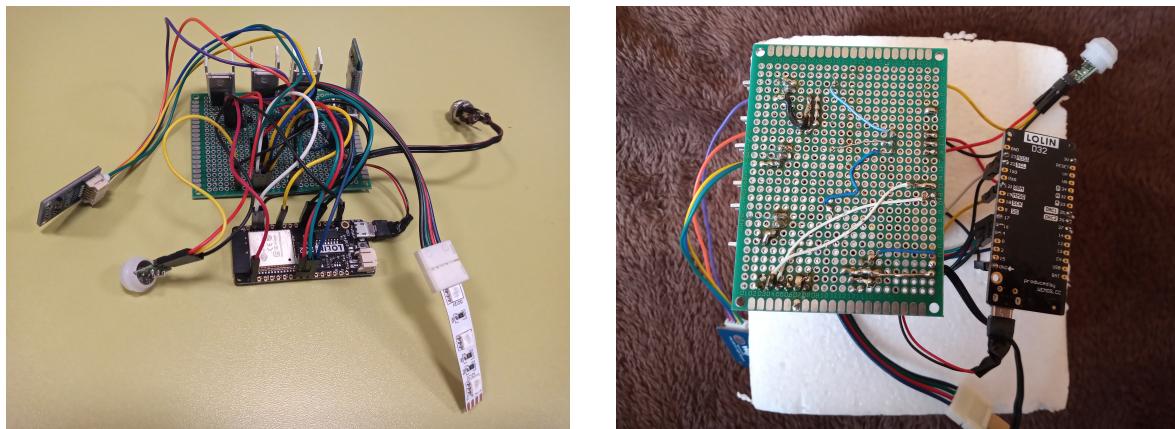
Postup zapnutia svetla pri detekcii pohybu je ilustrovaný na diagrame (Obr. 9).

Okrem rozhodnutia na pokyn zapnutia LED páiska sa želaná farebná teplota a jas lampy musí prepočítať na striedu pulzno-šírkovej modulácie pre zložky farby v RGB:

1. **Farebná teplota $[K]$ → RGB:** prevod podľa algoritmu využívajúceho aproximáciu závislosti cez lineárnu regresiu medzi farebnou teplotou a energiou vyžiarenu čiernym telesom v danom farebnom kanály podľa CIE 1964 colour-matching funkcie. Regresný model používaný algoritmom je dostatočne presný s R^2 nad 0,987 [10].
2. **RGB → HSV:** konverzia na model HSV pre úpravu jasu výslednej farby svetla
3. **$\mathbf{V} \leftarrow \text{Jas} [\text{lux}]$**
4. **HSV → RGB**
5. **PWM driver 8-bit výstup:** PWM kontroler má nastavené rozlíšenie presne na 8 bitov, čiže RGB trojica môže byť priamo nastavená do registra striedy výstupných kanálov.

3 Opis implementácie

Hardvérové súčiastky vnoreného systému sú osadené na univerzálnom plošnom spoji (Obr. 10). Keďže sa jedná o prototyp sú komponenty podľa možností pripojené cez ne-spájkované prepojovacie káblky alebo konektory, aby mohli byť jednoducho vymenené v prípade potreby. Na testovacie účely z dôvodu skladnosti bol využívaný jeden úsek LED pásku, ale v prevádzke sa nahradí metrovou hliníkovou lištou s matným krytom (Obr. 11). Lišta bude prilepená obojstrannou páskou na poličku nad stolom.



Obr. 10: Plošný spoj kontroléra inteligentného svietidla



Obr. 11: LED pásik namontovaný do metrovej hliníkovej lišty

Fimvér pre zariadenie kontroléra LED pásika na ESP32 je napísaný v jazyku C za použitia SDK **ESP-IDF** (Espressif IoT Development Framework) vo verzii 5.0 [2]. Ako vývojové prostredie sme použili *Visual Studio Code* a na zostavovanie a flashovanie programu slúži nástroj príkazového riadku *idf.py*.

Obsluhu činností ako sú napr. príjem príkazov cez Bluetooth alebo obsluha prerušenia od detektora pohybu zabezpečujú úlohy operačného systému *FreeRTOS*. Ovládače na bežné periférie ako sú: GPIO, PWM, Timer, UART, I2C, Non-volatile storage pochádzajú z SDK. Driver pre senzor osvetlenia od výrobcu pre vývojový platformu Arduino IDE je prepísaný a prispôsobený ku ESP-IDF.

```

1 typedef struct {
2     bool status;
3     bool movement;
4     uint8_t brightness;
5     uint16_t temperature;
6     uint16_t threshold;
7 } Light;
```

Zdrojový kód 1: Štruktúra vlastností svietidla

Aktívny stav svietidla je uložený v štruktúre Light, z ktorej existuje jedna globálna inštancia a prístup k nej je obmedzený cez mutex.

V hlavnom programe sa najprv načítajú nastavenia svetla od posledného vypnutia z nevolatilnej pamäte. Hoc by tento krok zlyhal použijú sa predvolené parametre (továrenské nastavenia). Ďalej sa nastaví interval pre časovač PIR senzora a obsluha prerušenia pri dosiahnutí času alarmu *switch_isr_handler*, ktorá dokáže notifikovať úlohu na vypnutie svetla *light_switch_task*.

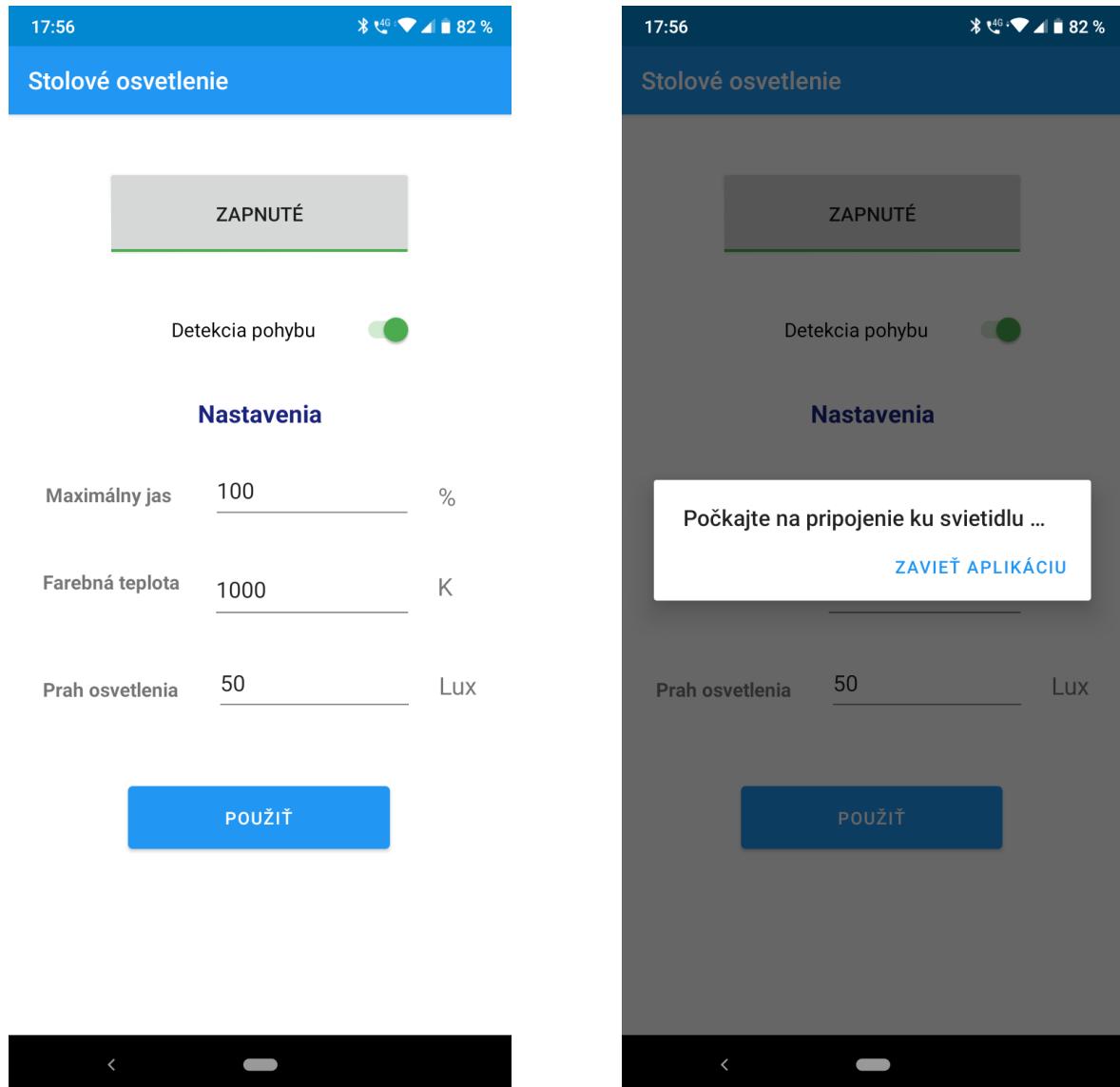
Z periférii je potrebné nastaviť zbernice pre komunikáciu so senzorom osvetlenia, Bluetooth modulom a nakoniec sa LED diódy uvedú do stavu podľa načítaných nastavení. Systém spúšta tri úlohy, ktoré prevažne blokujú pri čakaní na udalosti, bud' na požiadavku o zmenu stavu svetla alebo o asynchronný príjem príkazov na UART zbernicu.

```
1 void app_main(void)
2 {
3     light_mutex = xSemaphoreCreateMutex();
4     lux_sensor_mutex = xSemaphoreCreateMutex();
5
6     nvs_config();
7     nvs_load(&light);
8     timer_setup(&timer, PIR_TIMEOUT_S, timer_on_alarm);
9     pir_sensor_config(switch_isr_handler);
10    i2c_config();
11    light_sensor_config();
12    bluetooth_config();
13
14    led_config();
15    movement_detection(light.movement, timer, timer_on_alarm,
16                        switch_isr_handler);
16    uint16_t lux = illuminance_level();
17    lamp_scene(lux);
18
19    xTaskCreate(light_switch_task, "detect_movement", 2048, (void *)
20                true, 1, &gpio_task);
20    xTaskCreate(light_switch_task, "timeout_light", 2048, (void *)
21                false, 1, &timer_task);
21    xTaskCreate(receive_commands_task, "receive_commands", 4096, NULL,
22                10, NULL);
22 }
```

Zdrojový kód 2: Inicializácia hardvéru a spustenie úloh

Softvér mobilnej aplikácie je implementovaný v jazyku Kotlin vo vývojovom prostredí Android Studio v2022.1.1. Aplikácia podporuje ADK minimálne verzie 24, ale je zostavená a testovaná výhradne na smartfóne Motorola Moto E6 Plus s Android 9. Dizajn obrazovky je realizovaný podľa návrhu (Obr. 12). Pri spustení akejkoľvek akcie

sa overuje otvorené spojenie so svietidlom. Pokiaľ je dočasne nedostupné, dialógové okno zablokuje činnosť pre používateľa a snaží sa obnoviť Bluetooth pripojenie.



Obr. 12: Obrazovka mobilnej aplikácie

4 Otestovanie riešenia

Z dôvodu integrácie viacerých prvkov a konečnej použiteľnosti sme systém testovali cez **akceptačné testy**, ktoré definujú očakávané správanie z pohľadu používateľa. Vnorený systém prešiel všetkými uvedenými testami.

4.1 Test: Opakované nadväzovanie Bluetooth spojenia

Vstupné podmienky: Mobilná aplikácia ani lampa nie sú spustené. Bluetooth na smartfóne je vypnutý, ale HC-05 je už spárované.

Výstupné podmienky: Mobilná aplikácia zobrazuje nastavenia svetla.

Postup:

1. Používateľ otvorí aplikáciu s názvom „Stolové osvetlenie“ na svojom Android smartfóne
2. Aplikácia sa otvorí a zobrazí sa dialógové okno s vyžiadaním povolenia na aktivovanie Bluetooth.
3. Používateľ stlačí tlačidlo „Povolit“.
4. Aplikácia vypíše v dialógovom okne „Zapína sa rozhranie Bluetooth ...“, následne sa v smartfóne aktivuje Bluetooth. Aplikácia vypíše v dialógovom okne „Počkajte na pripojenie ku svietidlu ...“ a neumožňuje zrušenie tohto dialógového okna.
5. Používateľ pripojí lampu do elektrickej zásuvky a prepne prepínač lampy do polohy zapnuté.
6. Aplikácia maximálne do 15 sekúnd zatvorí dialógové okno a placeholders (vyšedené písmo) v textových poliach zmení za skutočné hodnoty (nevyšedené písmo).
7. Používateľ prepne prepínač lampy do polohy vypnuté a v mobilnej aplikácii klikne na ľubovoľné tlačidlo.
8. Aplikácie vypíše v dialógovom okne „Počkajte na pripojenie ku svietidlu ...“ a neumožňuje zrušenie dialógového okna.
9. Používateľ prepne prepínač lampy do polohy zapnuté.
10. Aplikácia znova maximálne do 15 sekund zatvorí dialógové okno a placeholders (vyšedené písmo) v textových poliach zmení za skutočné hodnoty (nevyšedené písmo).

4.2 Test: Aktívna detekcia pohybu

Vstupné podmienky: Bluetooth spojenie aplikácie a lampy je nadviazané. Senzor osvetlenia je úplne zatmavený. Detekcia pohybu je podľa prepínača v aplikácii vypnutá. Svetlo je vypnuté, aj podľa tlačidla v aplikácii, aj fyzicky.

Výstupné podmienky: Svetlo je vypnuté, aj podľa tlačidla v aplikácii, aj fyzicky.

Postup:

1. Používateľ stačí v aplikácii prepínač na zapnutie detekcie pohybu.
2. Aplikácia indikuje zmenou stavu prepínača v aplikácii, že detekcia pohybu je zapnutá.
3. Používateľ pohne rukou vo vzdialosti 1 m na priamu viditeľnosť od stredu šošovky PIR senzora.
4. Svetlo sa zapne, aj podľa tlačidla v aplikácii, aj fyzicky.
5. Používateľ počká bez akéhokoľvek okolitého pohybu po interval nastavený vo firmvéri, predvolene 20 s.
6. Svetlo sa po 20 s samočinne vypne, aj podľa tlačidla v aplikácii, aj fyzicky.

4.3 Test: Deaktivovaná detekcia pohybu

Vstupné podmienky: Bluetooth spojenie aplikácie a lampy je nadviazané. Senzor osvetlenia je úplne zatmavený. Detekcia pohybu je podľa prepínača v aplikácii vypnutá. Svetlo je vypnuté.

Výstupné podmienky: Svetlo je vypnuté, aj podľa tlačidla v aplikácii, aj fyzicky.

Postup:

1. Používateľ pohne rukou vo vzdialosti 1 m na priamu viditeľnosť od stredu šošovky PIR senzora.
2. Svetlo zostane vypnuté.

4.4 Test: Manuálne zapnutie a vypnutie svetla

Vstupné podmienky: Bluetooth spojenie aplikácie a lampy je nadviazané. Senzor osvetlenia je úplne zatmavený. Detekcia pohybu je podľa prepínača v aplikácii zapnutá, jas je 100%. Svetlo je vypnuté, aj podľa tlačidla v aplikácii, aj fyzicky.

Výstupné podmienky: Svetlo je vypnuté, aj podľa tlačidla v aplikácii, aj fyzicky.

Postup:

1. Používateľ stačí v aplikácii prepínač na vypnutie detekcie pohybu.

2. Aplikácia indikuje zmenou stavu prepínača v aplikácii, že detekcia pohybu je vypnutá.
3. Používateľ stačí v aplikácii tlačidlo „Vypnute”.
4. Svetlo sa zapne, aj podľa tlačidla v aplikácii, aj fyzicky.
5. Používateľ počká po interval nastavený vo firmvéri, predvolene min. 20 s.
6. Svetlo zostane zapnuté, aj po 30 s.
7. Používateľ stačí v aplikácii tlačidlo „Zapnute”.
8. Svetlo je vypnúté.

4.5 Test: Zmena farebnej teploty svetla

Vstupné podmienky: Bluetooth spojenie aplikácie a lampy je nadviazané. Senzor osvetlenia je úplne zatmavený. Svetlo je vypnuté, detekcia pohybu je vypnutá, jas je 100 %, farebná teplota je 3000 K.

Výstupné podmienky: Svetlo je zapnuté a svieti jasným bielym svetlom.

Postup:

1. Používateľ zadá do textového poľa „Farebná teplota” hodnotu „1000” a stlačí tlačidlo „Použit”.
2. Aplikácia akceptuje vstup a odošle farebnú teplotu cez Bluetooth na lampa.
3. Používateľ stačí v aplikácii tlačidlo „Vypnute”.
4. Svetlo sa zapne a svieti oranžovou farbou.
5. Používateľ zadá do textového poľa „Farebná teplota” hodnotu „5000” a stlačí tlačidlo „Použit”.
6. Aplikácia akceptuje vstup a svetlo zmení farbu na bielu.

4.6 Test: Zmena jasu svetla

Vstupné podmienky: Bluetooth spojenie aplikácie a lampy je nadviazané. Senzor osvetlenia je úplne zatmavený. Svetlo je vypnuté, detekcia pohybu je vypnutá, jas je 100%, farebná teplota je 3000 K.

Výstupné podmienky: Svetlo je zapnuté a svieti jasným bielym svetlom.

Postup:

1. Používateľ zadá do textového poľa „Maximálny jas” hodnotu „10” a stlačí tlačidlo „Použiť”.
2. Aplikácia akceptuje vstup a odošle jas cez Bluetooth na lampu.
3. Používateľ stačí v aplikácii tlačidlo „Vypnute”.
4. Svetlo sa zapne a svieti bielou farbou slabej intenzity.
5. Používateľ zadá do textového poľa „Maximálny jas” hodnotu „0” a stlačí tlačidlo „Použiť”.
6. Svetlo je zapnuté, ale nesvieti.
7. Používateľ zadá do textového poľa „Maximálny jas” hodnotu „100” a stlačí tlačidlo „Použiť”.
8. Svetlo svieti bielou farbou silnej intenzity.

4.7 Test: Reakcia svetla na okolité osvetlenie

Vstupné podmienky: Bluetooth spojenie aplikácie a lampy je nadviazané. Senzor osvetlenia odkrytý a zo vzdialenosťi 10 cm svieti naň „Baterka” zo smartfónu. Svetlo je vypnuté, detekcia pohybu je vypnutá, intenzita je 100%. **Výstupné podmienky:** Svetlo je zapnuté.

Postup:

1. Používateľ zadá do textového poľa „Prah osvetlenia” hodnotu „10” a stlačí tlačidlo „Použiť”.
2. Aplikácia akceptuje vstup a odošle prahovú hodnotu cez Bluetooth na lampu.
3. Používateľ stačí v aplikácii tlačidlo „Vypnute”, to zmení nápis na „Zapnuté”.
4. Svetlo sa nezapne a text tlačidla sa vráti naspäť na „Vypnute”
5. Senzor osvetlenia je zatmavený a prestane naň svietiť „Baterka”. Používateľ stačí v aplikácii tlačidlo „Vypnute”, to zmení nápis na „Zapnuté”.
6. Svetlo sa zapne.

4.8 Test: Načítanie pôvodných nastavení po zapnutí

Vstupné podmienky: Bluetooth spojenie aplikácie a lampy je nadviazané. Senzor osvetlenia je úplne zatmavený. Svetlo je zapnuté, detekcia pohybu je vypnutá. **Vý-**

stupné podmienky: Svetlo je vypnuté, detekcia pohybu je vypnutá, jas je 100%, farebná teplota je 5000 K, prah je 1000.

Postup:

1. Používateľ zadá do textového polí hodnoty: „Farebná teplota“ = „5000“, „Maximálny jas“ = „100“, „Prah osvetlenia“ = „1000“ a stlačí tlačidlo „Použit“.
2. Aplikácia akceptuje vstup a odošle hodnoty cez Bluetooth na lampu.
3. Svetlo sa bude mať vlastnosti podľa zmenených nastavení.
4. Používateľ stačí v aplikácii tlačidlo „Zapnuté“, to zmení nápis na „Vypnuté“.
5. Svetlo sa vypne.
6. Používateľ prepne prepínač lampy do polohy vypnuté a zatvorí mobilnú aplikáciu, počká na zhasnutie indikačnej ledky na MCU.
7. Používateľ prepne prepínač lampy do polohy zapnuté a otvorí mobilnú aplikáciu.
8. Aplikácia sa pripojí na lampu cez Bluetooth a načíta posledné uložené nastavenia.
9. Používateľ stačí v aplikácii tlačidlo „Vypnute“.
10. Fyzické vlastnosti svetla budú rovnaké ako pred vypnutím.

5 Zhodnotenie výsledkov

V semestrálnom projekte sme vytvorili vnorený systém postavený na vývojovej doske ESP32 Lolin D32, podľa zadania a špecifikácie požiadaviek. Zariadenie ovláda nastavenia RGB LED pásku podľa vstupov od PIR senzora pohyba, senzora vonkajšieho osvetlenia, a príkazov z mobilnej aplikácie na platforme Android komunikujúcej cez Bluetooth RFCOMM s modulom HC-05.

Na základe analýzy dostupnej ponuky vhodných typov senzorov a LED osvetlenia sme navrhli systém na spínanie stolnej lampy, pričom sme museli vyriešiť distribúciu viacerých napájacích napäť pre prvky v systéme, pripojenie a komunikáciu so senzormi cez mikroprocesorové zbernice, a scenáre spínania svetla. Firmvér je implementovaný v jazyku C v SDK ESP-IDF. Mobilná aplikácia je napísaná v jazyku Kotlin a jej hlavnou úlohou je posielanie Bluetooth príkazov zariadeniu.

Správnosť systému ako celku sme overili sériou akceptačných testov s kladnými výsledkami pri každom z nich. Použitie RGB svetiel umožňuje rozšírenie do budúcnosti pre ďalšie farebné scény.

Zoznam použitej literatúry

1. *ESP32-WROOM-32 Datasheet*. Espressif Systems, 2023. Dostupné tiež z: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.
2. *ESP-IDF Programming Guide* [online]. [cit. 2023-04-13]. Dostupné z : <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/index.html>.
3. MISCHIANTI, Renzo. *ESP32 WeMos LOLIN D32 high resolution pinout and specs* [Renzo Mischianti] [online]. 2023-02-01. [cit. 2023-04-13]. Dostupné z : <https://www.mischianti.org/2023/02/01/esp32-wemos-lolin-d32-high-resolution-pinout-and-specs/>. Section: WeMos LOLIN D32.
4. *HC-05 - Bluetooth Module* [Components101] [online]. [cit. 2023-04-13]. Dostupné z : <https://components101.com/wireless/hc-05-bluetooth-module>.
5. 123LED.SK. *Ako vybrať LED pásik* [123led.sk] [online]. [cit. 2023-04-13]. Dostupné z : <https://www.123led.sk/ako-vybrat-led-pasik/>.
6. 123LED.SK. *RGB LED pásik 14,4W/m 12V bez krytia IP20* [123led.sk] [online]. [cit. 2023-04-13]. Dostupné z : <https://www.123led.sk/rgb-led-pasik-14-4w-m-bez-krytia/>.
7. *BD711 / BD712 - Complementary Silicon Power Transistors*. ARK Component, [b.r.].
8. OLIMEX. *PIR-SB312* [Olimex] [online]. [cit. 2023-04-13]. Dostupné z : <https://www.olimex.com/Products/Components/Sensors/PIR-SB312/>.
9. *TSL25911 Light Sensor - Waveshare Wiki* [online]. [cit. 2023-04-13]. Dostupné z : https://www.waveshare.com/wiki/TSL25911_Light_Sensor.

10. HELLAND, Tanner. *How to Convert Temperature (K) to RGB: Algorithm and Sample Code* [tannerhelland.com] [online]. 2012-09-18. [cit. 2023-04-13]. Dostupné z : <https://tannerhelland.com/2012/09/18/convert-temperature-rgb-algorithm-code.html>.

6 Technická dokumentácia

6.1 Prehľad hardvérových súčiastok

Na základe predošej analýzy sme zostavili zoznam súčiastok potrebných pre zhotovenie zariadenia. Cena hardvéru prototypu (jednokusová výroba) je odhadovaná na 55 €, na základe nami zvolených dodávateľov.

Riadiaca jednotka

- ESP32 WeMos Lolin D32 - Mikrokontrolér
- HC-05 - Bluetooth modul

Senzory a akčné členy

- RGB LED pásik 14,4W/m 12V bez krytia IP20 1m
- Olimex PIR-SB312 10x8mm
- WS-17146 TSL25911 Light Sensor

Elektrické súčiastky

- L7805CV Lineárny regulátor napäťia 12V na 5V
- NPN Tranzistor BD711
- Rezistory - 220 Ω , 330 Ω
- LED zdroj (trafo) 12V 30W IP67
- Flexo šnúra – 3m
- Konektory: RGB LED pásik, Micro USB-B, DC konektor a zásuvka

Mechanické súčiastky

- DO1 chladič
- DO3A chladič
- Univerzálny plošný spoj
- Nástenný profil N3 biely, Opálový kryt 1m
- Koncovka profilu N3 biela
- Vypínač mezišnúrový

6.2 Zdrojové súbory pre firmvér

Implementácia firmvéru sa nachádza v priečinku *esp*. Na nahratie firmvéru na MCU je potrebné mať stiahnuté a nainštalované ESP-IDF s požadovanými systémovými závislostami podľa dokumentácie. Potom vykonáme v priečinku *esp* tieto príkazy:

```
source /path/to/esp-idf/export.sh  
idf.py build  
idf.py -p /dev/ttyUSB0 flash
```

Firmvér je rozdelený do viacerých súborov zdrojového kódu obsahujúce popísanú funkciuonalitu:

- **main.c** - Hlavný program, úlohy a obsluhy prerušení.
- **storage.c** - Načítanie a zapis nastavení svetla z/do nevolatilnej pamäte.
- **trigger.c** - Nastavenie časovača a prerušení po timeout a pri detekcii pohybu.
- **i2c.c** - Zjednodušenie rozhrania pre komunikáciu cez I2C zbernicu.
- **lightsensor.c** - Ovládač pre senzor osvetlenia. Externe prístupné sú iba funkcie *light_sensor_config()*, *light_sensor_read_lux()*
- **bluetooth.c** - Inicializácia UART linky na príslušných pinoch *bluetooth_config()*, parsovanie príkazov v priatej správe na ovládanie svetla *parse_commands()* a odoslanie aktuálneho stavu lampy *bluetooth_send_status()*
- **led.c** - Konfigurácia PWM kanálov *led_config()* a nastavenie LED diód podľa rôznych parametrov, buď cez priamo požadovanú RGB farbu *led_set_color()* alebo s prepočtom z farby a jasu *led_output()*.

6.3 Zdrojové súbory pre Android aplikáciu

Zdrojový kód mobilnej aplikácie a nastavenia vývojového prostredia sa nachádzajú v priečinku *led-light-app*, s rozmiestnením súborov podľa predvolenej štruktúry z Android Studio. Hlavná aktivita sa nachádza v súbore *MainActivity.kt* vnútri *app/src/main/java/com/smartdevice/tablelantern/*. Rozmiestňuje UI prvky na obrazovke, iniciuje a prijíma správy od Bluetooth služby v *BluetoothService.kt*.

V triedach prítomných v aplikácii sa nachádzajú tieto najdôležitejšie metódy:

class MainActivity

- `onCreate()` - vytvorenie UI komponentov a inicializácia Bluetooth služby (*BluetoothService*).
- `onSwitchLight()` - Udalosť tlačidla na odoslanie príkazu cez Bluetooth na zmenenie stavu svetla.
- `onMotionDetectSwitch()` - Udalosť tlačidla na odoslanie príkazu cez Bluetooth na (de)aktiváciu detektora pohybu.
- `onApplySettings()` - Udalosť tlačidla na odoslanie želaných nastavení svetla s ich validáciou.
- `lampConnect()` - Nájdenie modulu HC-05 medzi spárovanými zariadeniami a otvorenie Bluetooth spojenia, na ten účel volá metódu `BluetoothService.connect()`
- `setLightState()` - Spätné volanie (callback) po prijatí konfigurácie od vno- reného systému na jej parsovanie a načítanie do textových polí v aplikácii.
- `BlHandler.handleMessage()` - Zobrazovanie správ z Bluetooth služby v UI rozhraní.

class BluetoothService

- `connect()` - Spúšta vlákno *ConnectThread*. Slúži ako rozhranie pre *MainActivity* na prístup k vnútorným triedam vlákin.
- `connected()` - Spúšta vlákno *ConnectThread*. Slúži ako rozhranie pre *MainActivity* na prístup k vnútorným triedam vlákin.
- `send()` - Služí ako rozhranie voči *MainActivity* a volá `ConnectedThread.write()` so argumentmi v správnom formáte.

- `ConnectThread.run()` - Hlavná metóda vlákna *ConnectThread* na nadviazanie Bluetooth spojenia cez RFCOMM socket. Volá metódu *connected*.
- `ConnectedThread.run()` - Hlavná metóda vlákna *ConnectedThread* čaká na príjem bajtov z otvoreného Bluetooth socket. Po prijatí celistvej správy stavu vnoreného systému notifikuje hlavnú aktivitu.
- `ConnectedThread.write()` - Metóda na odoslanie bajtov na zariadenie cez otvorený Bluetooth socket.

6.4 Syntax Bluetooth príkazov

Zariadenie lampy komunikuje cez Bluetooth RFCOMM prostredníctvom preddefinovanej sady textových príkazov. Príkazy posielané hneď za sebou musia byť oddelené ľudovoľným oddelovačom napr. medzerou alebo čiarkou. Po poslaní aktivovaní konfiguračného príkazu sa ako odpoveď vráti aktuálny stav svetla vo formáte podľa príkazu *REQ*.

- **SWITCH** - Prenutie aktuálneho stavu svetla. Ak je zapnuté, tak sa vypne a naopak.
- **DETECT** - Prenutie aktuálnej aktívnosti detektora pohybu.
- **LEVEL [level]** - Jas svetla v percentách z celkovej svietivosti. Validná hodnota pre parameter *level* je celé číslo v rozsahu 0 - 100 vrátane.
- **KELVIN [temp]** - Farebná teplota svetla v kelvinoch. Validná hodnota pre parameter *temp* je celé číslo v rozsahu 1000 - 40000 vrátane.
- **LUX [threshold]** - Prahová úroveň vonkajšieho osvetlenia v luxoch, pod ktorú sa svetlo rozsvieti. Validná hodnota pre parameter *threshold* je celé číslo v rozsahu 0 - 10000 vrátane. Nepredpokladá sa zapínanie svetla pri vysokej úrovni vonkajšieho osvetlenia.
- **REQ** - Aktuálny stav svetla vo formáte: „*S/0,M/0,K/2000,B/100,L/5*“. Písmená označujú príslušné parametre takto: **S** - stav svetla (0 alebo 1), **M** - aktívnosť detektora pohybu (0 alebo 1), **K** - farebná teplota, **B** - jas, **L** - prahová úroveň osvetlenia. Rozsah hodnôt je rovnaký ako pri konfiguračných príkazov.