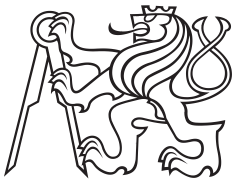
Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra radioelektroniky

RGB LED světlo se vzdáleným ovládáním

Filip Křemen

Vedoucí: doc. Ing. Stanislav Vítek, Ph.D.
Zaměření: Elektronika a komunikace
Květen 2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Křemen** Jméno: **Filip** Osobní číslo: **474717**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra radioelektroniky**

Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

RGB LED světlo se vzdáleným ovládáním

Název bakalářské práce anglicky:

RGB LED Light with Remote Control

Pokyny pro vypracování:

Navrhněte a implementujte RGB LED světlo se vzdáleným ovládáním. Driver pro výkonovou RGB LED diodu umožní nezávislé ovládání barevných kanálů. Světlo bude možné ovládat pomocí bezdrátové osobní sítě WPAN. Navrhněte komunikační protokol, který umožní nastavení světelného toku a barvy světla. Vlastnosti světla ověřte vhodnou měřicí metodou, např. spektrofotometrem.

Seznam doporučené literatury:

[1] FALUDI, Robert. Building wireless sensor networks: with ZigBee, XBee, arduino, and processing. " O'Reilly Media, Inc.", 2010.
[2] SERPANOS, Dimitrios; WOLF, Marilyn. Internet-of-things (IoT) systems: architectures, algorithms, methodologies. Springer, 2017.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Stanislav Vítek, Ph.D. katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **01.02.2022** Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **30.09.2023**

doc. Ing. Stanislav Vítek, Ph.D.	doc. Ing. Stanislav Vítek, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. _____ Datum převzetí zadání

_____ Podpis studenta

© ČVUT v Praze, Design: ČVUT v Praze, VIC

Poděkování

Chtěl bych poděkovat panu doc. Ing. Stanislavu Vítkovi, Ph.D. za trpělivost a komunikaci při vypracování této práce. Děkuji také rodině, přátelům za podporu a firmě STMicroelectronics za poskytnutí Nucleo kitu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 19. května 2022

Abstrakt

Práce se zabývá navržením RGB LED driveru a následným dálkovým ovládáním pomocí modulu Xbee. V úvodu jsou popsané IoT systémy a jejich protokoly. Dále jsou v práci vysvětleny principy PWM modulace při řízení LED. Následuje návrh řídicí desky a komunikačního protokolu. V závěru je pak měření vlastností RGB LED na spektrofotometru a zhodnocení navrženého systému.

Klíčová slova: RGB LED, PWM, STM32, IoT, ZigBee

Vedoucí: doc. Ing. Stanislav Vítek, Ph.D.

Abstract

This thesis deals with the design of RGB LED driver and with remote control using the Xbee module. IoT systems and their protocols are described in the introduction. Furthermore, the thesis explains the principles of PWM modulation for controlling RGB LED. This is followed by the design of control board and the communication protocol. Finally, it concludes with the measurement of RGB LED properties on a spectrophotometer and evaluation of the proposed system.

Keywords: RGB LED, PWM, STM32, IoT, ZigBee

Obsah

Obrázky

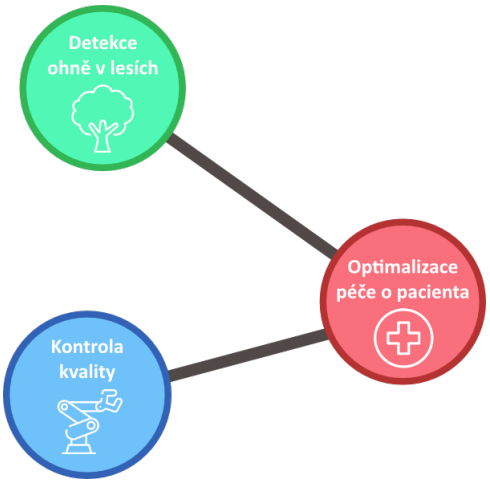
Tabulky

Kapitola 1

Úvod

1.1 Internet věcí - Internet of Things

Internet věcí patří v poslední době k nejnovějším trendům na poli informačních technologií. IoT systém tvoří typicky více zařízení, která jsou vzájemně propojena a tvoří tak komplexní celek. Primárně se jedná o získávání dat ze senzorů, kde je cílem data například zaslat na vzdálený server a tam je dále zpracovávat. Takový systém ale nemusí obsahovat jen senzory, součástí můžou být i aktuátory. Může se jednat o senzorovu síť pro monitorování, zabezpečovací systém, kde v případě nějaké údalosti přijaté ze senzoru dojde ke spuštění alarmu a nebo o řídící systém, kde jsou data ze senzoru zasílána jako vstup do řídicích algoritmů pro ovládání aktuátorů.



Obrázek 1.1: Příklady IoT systémů

IoT systémy nalézají využití také v průmyslu při kontrole produktů ve výrobě. V moderních budovách se pak může jednat o sledovaní teploty v

místnostech (ovládání klimatizace) nebo můžeme získávat přehled o stavu budovy. Díky rozvoji MEMS senzorů lze vyvíjet zařízení IoT levně a v malých rozměrech, což usnadňuje integraci do našeho systému. Důležitým bodem v návrhu systému je spotřeba daného řešení a také zabezpečení dat, která posíláme mezi zařízeními. [1]

1.2 Protokoly pro IoT systémy

Existuje řada protokolů sloužících ke komunikaci mezi zařízeními. Naše požadavky můžou být různé, je-li zařízení napájeno z akumulátoru o nízké kapacitě, je nutné zvolit takový protokol, který nebude moc energeticky náročný a přenosová rychlost pro sběr dat bude dostatečná. Budeme-li potřebovat zasílat data na velké vzdálenosti, musíme zvolit jinou technologii. Dalším aspektem může taktéž být spolehlivost přenosu dat. Při měření vlhkosti vzduchu v rodinném domě ničemu neuškodí, když se nějaká hodnota při přenosu ztratí, problém by ale mohl být někde v průmyslu, kde by špatná hodnota, například o vzdálenosti, mohla mít fatální následky. Nyní si jednotlivé protokoly používané pro IoT blíže představíme.

1.2.1 Bluetooth

Obrázek 1.2: Bluetooth [2]

Technologie Bluetooth vznikla jako náhrada kabelového připojení pro periferie nebo pro přenos dat. Existují dvě varianty - Bluetooth Classic a Bluetooth Low Energy. Bluetooth Classic nalezneme v bezdrátových reproduktorech nebo headsetech. Od verze 4.0 přišlo Bluetooth Low Energy, které je uzpůsobené tak, aby mělo co nejmenší spotřebu, je tedy vhodné do zařízení na akumulátory s malou kapacitou. Nevýhodou je, že zařízení s verzí Classic nemůže přímo komunikovat se zařízením Low Energy, proto některá zařízení mají implementovány obě technologie a pracují v tzv. duálním módu. Typická maximální vzdálenost pro přenos přes BLE je 10 až 30 metrů. Špičkový proud při přenosu je typicky pod 15 mA. Níže je tabulka popisující rozdíly mezi těmito dvěma variantami. Obě verze pracují v pásmu 2.4 GHz. [3]

Bluetooth Classic	BLE
Používá se pro streamování hudby přenos souborů a headsety	Používá se pro sběr dat ze senzorů, ovládání zařízení
Chybí přizpůsobení pro malou spotřebu, ale má větší přenosovou rychlost (3 Mbps, BLE má 2 Mbps)	Uzpůsobeno pro malou spotřebu vhodné pro sběr dat s nízkou frekvencí
Pracuje s více než 79 kanály	Pracuje s více než 40 kanály
Vyhledávání ve 32 kanálech	Vyhledávání ve 3 kanálech rychlejší nalezení a připojení

Tabulka 1.1: Porovnání Bluetooth Classic a BLE [3]

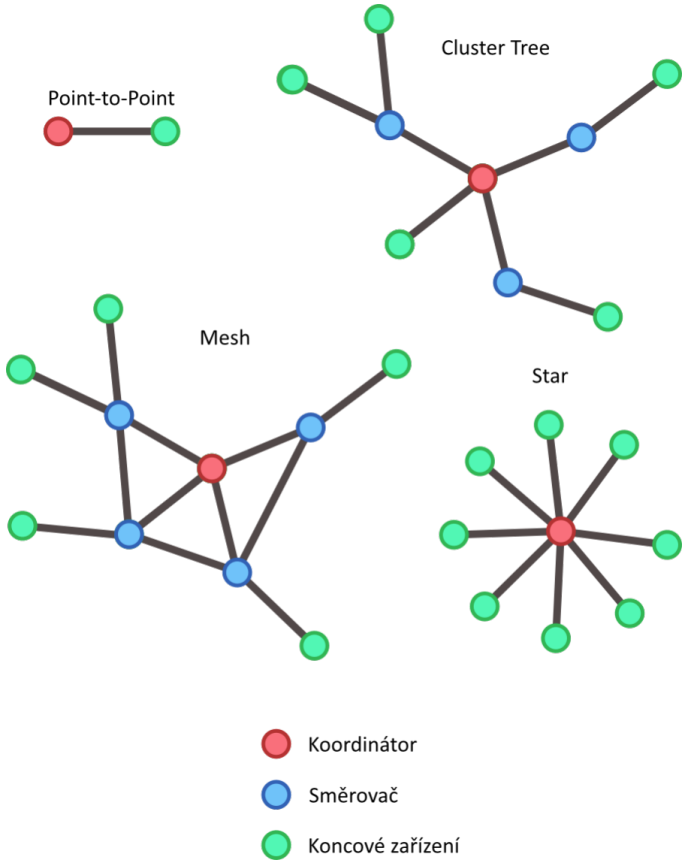
1.2.2 ZigBee

Obrázek 1.3: ZigBee [4]

Protokol ZigBee je protokol založený na standardu IEEE 802.15.4 pro vytváření bezdrátových sítí WPAN - wireless personal area networks. Tento protokol se uplatní v aplikacích s nízkou spotřebou a malou šířkou pásma. Typické použití nalezneme v domácí automatizaci, v průmyslu pro řídicí systémy nebo v medicínské oblasti pro sběr dat. Rychlost přenosu je 250 kbit/s pro pásmo 2.4 GHz a to typicky do 10 až 20 metrů. Síť protokolu ZigBee obsahuje koordinátora, směrovač a koncové zařízení. Koordinátor je v každé síti pouze jeden, stará se o chod celé sítě a přidělování adres. Dále je tu směrovač, jehož úkolem je propojovat zařízení mezi sebou (např. koordinátor-směrovač-koncové zařízení). Většinou jsou tato zařízení napájena ze sítě, aby byl zajištěn nepřetržitý chod. Koncová zařízení se pak můžou připojit ke koordinátoru nebo směrovači a můžou pouze posílat a přijímat zprávy ze sítě. Každé rádio ZigBee má přidělené svoje 64-bitové sériové číslo, které je unikátní. V každé síti pak má dané zařízení kratší 16-bitovou dynamicky přiřazovanou adresu a je unikátní pouze v dané síti. Dále každý komunikační uzel může mít svůj identifikátor - jméno, což je praktické pro lepší přehlednost. Každá vytvořená síť má také svoji adresu a to 16-bitové číslo. Jedná se o PAN adresu (Personal Area Network). K fungování sítě je nutné také nastavit komunikující rádia na stejný kanál. Každé zařízení v jedné síti musí mít nastavený stejný kanál. [5] Na obrázcích níže je ukázka adres a jednotlivé topologie ZigBee.

Typ	Příklad	Unikátnost
64-bitová	0013A200403E0750	Vždy a všude
16-bitová	23F7	Jen v jedné síti
Identifikátor uzlu	senzortlaku	Není zaručena

Tabulka 1.2: Adresy ZigBee [5]



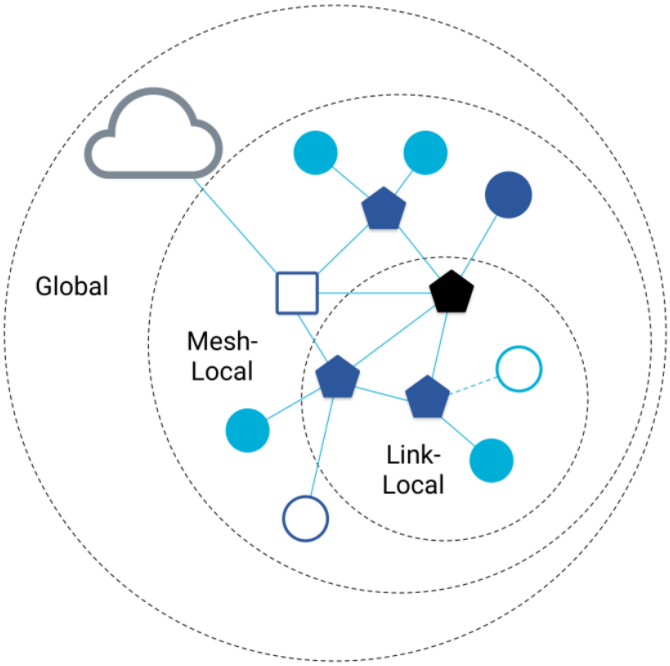
Obrázek 1.4: Typy topologie sítě ZigBee - inspirováno z [5]

1.2.3 Thread

Obrázek 1.5: Thread [6]

Dalším protokolem je Thread, který je založený na IPv6 a určený pro WPAN sítě. Mezi jeho výhody patří snadná instalace a nízká energetická náročnost. Všechna zařízení jsou v síti autentifikována a veškerá komunikace

je šifrovaná. Samotná síť může obsahovat stovky zařízení. Podobně jako u ZigBee protokolu má Thread síť směrovače a koncová zařízení, s tím rozdílem, že koordinátor tu není. Směrovač se stará o přeposílání paketu po síti a o připojování dalších zařízení. Koncová zařízení komunikují pouze se směrovači, nepřeposílají žádné pakety a mohou se uspat pro snížení spotřeby. Thread protokol má tři úrovně pro adresování. První je úroveň lokální - jednotlivá zařízení vídí jen své sousední zařízení, následuje síťová, kdy na sebe jednotlivá zařízení vidí v rámci stejné Thread síti. Poslední úroveň je globální, kde na sebe zařízení vídí i mimo Thread síť. Lokální a síťová úroveň mají prefix $fe80 :: /16$ a fd00::/8. [7]



Obrázek 1.6: Úrovně adresování protokolu Thread [7]

1.2.4 LoRa WAN

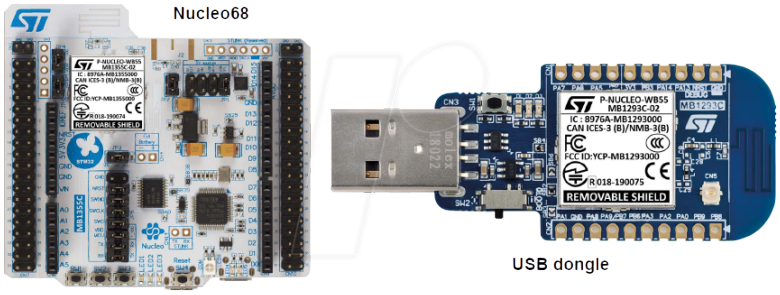
LoRaWAN je Media Acces Control protokol postavený na LoRa modulaci, která je založená na technologii Chirp Spread Spectrum. Jedná se o softwarovou vrstvu, která definuje to, jak mají zařízení využívat LoRa hardware. Komunikace postavená na LoRaWAN je vhodná pro sběr dat o malých objemech na dlouhé vzdálenosti 3 až 10 km. LoRaWAN poskytuje optimalizaci, tak aby dané zařízení vydrželo až 10 let na knoflíkové baterii. Výhodou je, že není třeba platit poplatky za využívání pásma, protože protokol umožňuje pracovat v bezlicenčních pásmech (915 MHz, 868 MHz a 433 MHz). Dále se může využívat 2.4 GHz pásmo k dosažení vyšších přenosových rychlostí ale za cenu snížení dosahu. Taktéž protokol umožňuje geolokaci koncového zařízení pomocí triangulace, jen je nutné, aby alespoň 3 vstupní brány zachytily jeho

Obrázek 1.7: LoRa WAN [8]

signál. Dále nabízí end-to-end zabezpečení komunikace a také dobré šíření signálu i uvnitř budov. [9]

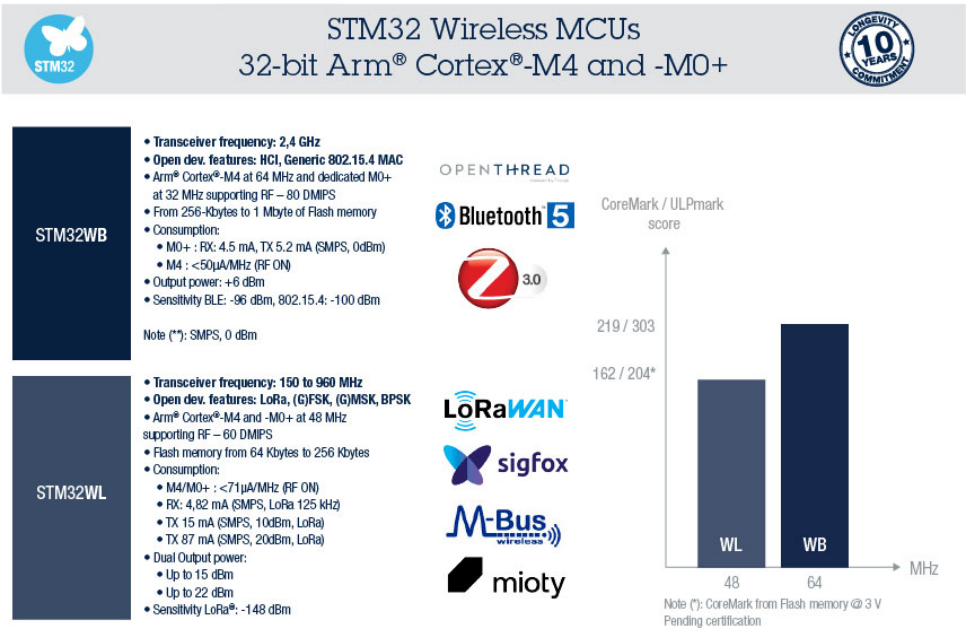
1.3 Mikrokontroléry

Data ze senzorů musíme nějakým způsobem získat a k tomu slouží například mikrokontroléry. Jedná se o mikropočítač, který má paměti a periferie zabudované už v pouzdře. Data ze senzorů můžeme získat AD převodníkem (převodem analogové hodnoty na digitální) nebo, pokud to již daný senzor nabízí, využít komunikační sběrnice UART, I2C, SPI případně v automobilovém průmyslu CAN. V současné době je hodně rozšířené v hobby komunitě Arduino, díky němuž se dá do světa mikrokontrolérů proniknout rychle. Okolo Arduino je veliká komunita, existuje mnoho materiálu, a proto je Arduino tak oblíbené. Další velice populární vývojovou deskou je ESP32, respektive ESP866, které oproti základnímu Arduino Uno nabízí konektivitu pomocí Bluetooth nebo Wi-Fi. Nevýhodou Arduino je jeho cena vzhledem k jeho možnostem, ta je ale na druhou stranu vykoupena skvělou podporou. Často se obchází tento nedostatek koupí Arduino v čínském e-shopu (Arduino je open-source projekt), zde ale může být problém s instalací ovladačů na počítači. Uveďme si příklady některých vývojových kitů pro IoT. Firma STMicroelectronics nabízí ve svém portfoliu dvě série bezdrátových mikrokontrolérů - STM32WB a STM32WL. Série STM32WB podporuje Bluetooth Low Energy ve verzi 5.2, komunikační protokoly IEEE 802.15.4 Zigbee a Thread. Protokoly mohou běžet samostatně nebo souběžně.



Obrázek 1.8: Vývojový kit NUCLEO-WB55 [10]

STM32WL série umožňuje multimodulaci, proto je vhodná pro vývoj IoT systémů s velmi nízkou spotřebou při zachování výkonu. Podporuje protokoly LoRa, Sigfox a M-Bus. [11]



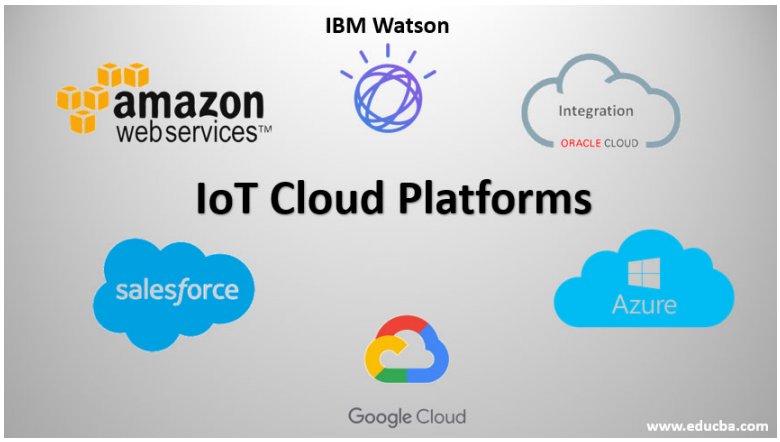
Obrázek 1.9: Porovnání sérií STM32WB a STM32WL [11]

U těchto mikrokontrolérů je vše zabudované již v pouzdru. Pokud chceme připojit mikrokontrolér k naší aplikaci pomocí Bluetooth, ale daný mikrokontrolér Bluetooth nepodporuje, je nutné připojit externí modul, se kterým bude mikrokontrolér komunikovat pomocí výše zmíněných sběrnic. Společnost Digi vytvořila ekosystém Digi Xbee, který obsahuje moduly, knihovny pro vývoj a další vývojové prostředky. Pro nízkopříkonové, point-to-point a mesh sítě nabízí společnost připojení pomocí modulů nabízejících Zigbee, DigiMesh, Bluetooth a protokoly z IEEE 802.15.4. K vývoji se dají využít existující knihovny pro jazyky Python, Ansi C, Java a další. Níže je uvedená vývojová sada použitá v této práci, modul Xbee nabízí tzv. transparentní mód, který je pro tuto aplikaci vhodný, více si o tomto módu povíme v dalších kapitolách.

Obrázek 1.10: Vývojová sada Xbee Wireless Connectivity Kit [12]

1.4 Zpracování dat a cloudové služby

Pokud máme již IoT systém navržený, tak se nabízí získaná data ze senzoru ještě dále zpracovat. Taková data se dají použít pro predikování vývoje systému nebo měřené veličiny. S využitím strojového učení lze pomocí klasifikátoru zjistit, v jakém stavu se systém nachází nebo jak se daná veličina bude v čase měnit. Pro takové algoritmy je zapotřebí nasbírat soubor dat. Nasbíraná data je možno uložit lokálně na svém počítači nebo využít některé z cloudových služeb uzpůsobených pro IoT systémy. Pomocí těchto služeb máme o datech podrobný přehled a můžeme s nimi lépe pracovat.



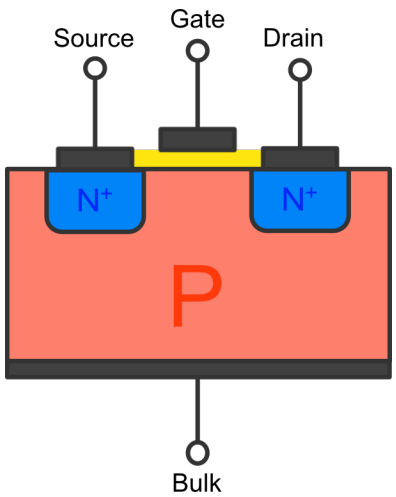
Obrázek 1.11: Cloudové služby pro IoT [13]

Kapitola 2

Princip řízení RGB LED

2.1 MOSFET tranzistor

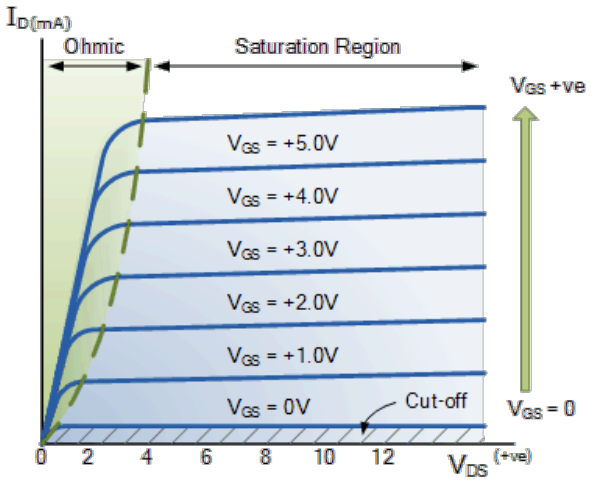
Tranzistor MOSFET (metal oxide field effected transistor) je polovodičová součástka, která je řízená elektrickým polem - napětím. Může sloužit jako spínač nebo jako řízený zdroj proudu pomocí elektrického napětí. Jeho hlavní část se skládá z polovodiče typu N a nebo P. Pokud je tranzistor typu N-kanál, jeho hlavní část je z polovodiče P a dvě oblasti z vysoce dotovaného polovodiče typu N+ (v případě P-kanálu je to naopak). Dále se rozlišuje indukovaný a nebo zabudovaný vodivostní kanál.



Obrázek 2.1: Průřez MOSFET tranzistoru (N-kanál) - inspirováno z [14]

Žlutá oblast na obrázku mezi elektrodou Gate a P oblastí tvoří oxid křemíku, který tvoří dielektrikum. Elektrody Source a Bulk bývají propojeny. Řídící napětí se pohybuje typicky ±20 V, kdy při překročení této hodnoty dojde

k průrazu oxidu a zničení tranzistoru. Je důležité si pohlídat tuto hodnotu, jestliže používáme nějaký integrovaný MOSFET driver, může se stát, že výstupní napětí takového obvodu je vyšší než dovolené napětí U_{GS}, vždy je nutné se dívat do datasheetu dané součástky. Princip MOSFETu (N-kanál) je následovný, přiložením kladného napětí na elektrody Gate a Source dojde k vytvoření vodivého kanálu pod oxidem křemíku. Elektrony z P oblasti se shromáždí u kladného potenciálu Gate elektrody. Tímto vznikne vodivá cesta mezi elektrodou Source a Drain. Měníme-li napětí Ugs, měníme tím i maximální proud Drainem I_D. Jelikož se součástka řídí elektrickým napětím, je vhodná pro použití s mikrokontroléry pro řízení výkonu.[15]



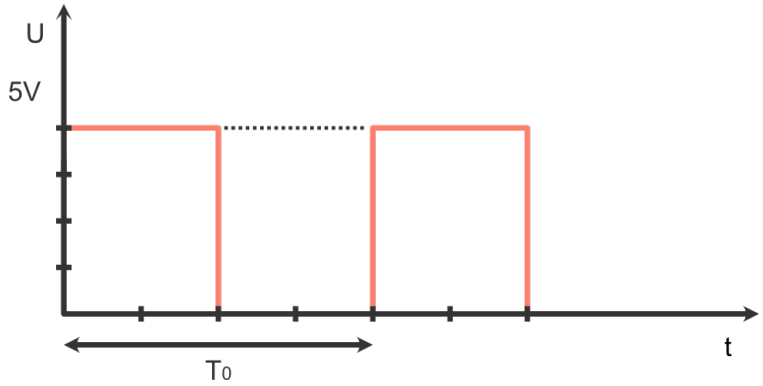
Obrázek 2.2: Výstupní charakteristiky N-Mosfetu [16]

2.2 PWM modulace

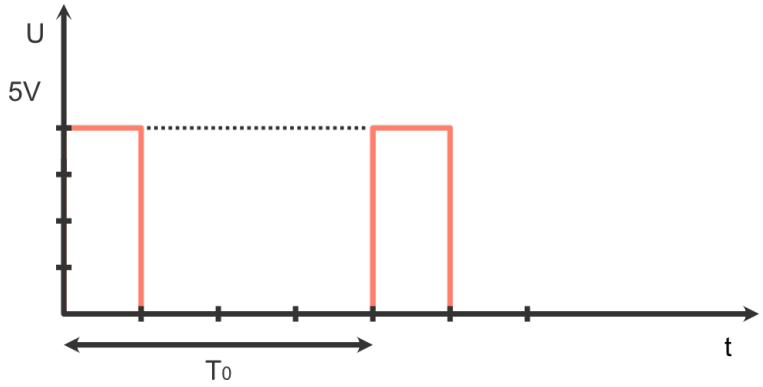
Pulzně šířková modulace (pulse width modulation) se používá v řadě aplikací. Využít ji lze například ke generování sinusového signálu nebo k řízení výkonu. Máme-li obdelníkový signál se základní periodou T_0, ve které měníme poměr stavů *zapnuto* a *vypnuto*. Střída signálu je pak dána vzorcem

$$\text{Duty Cycle} = \frac{T_{\text{on}}}{T_{\text{off}}} * 100\ [\%]. \tag{2.1}$$

Tímto způsobem měníme výkon na každém barevném kanálu LED světla a tím měníme barvu (změnou intenzity jednotlivých kanálů se nám jeví daná barva) či jas. Níže jsou uvedené příklady se střídou 50% a 25%.

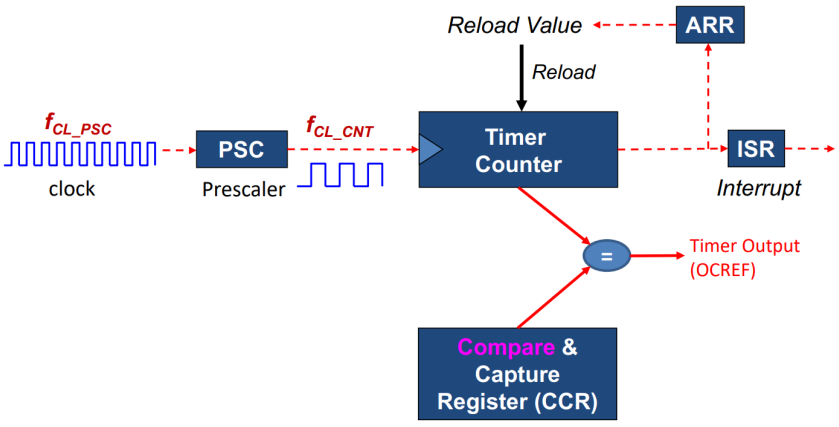


Obrázek 2.3: PWM modulace - 50% střída



Obrázek 2.4: PWM modulace - 25% střída

2.2.1 Generování PWM modulace pomocí mikrokontroléru



Obrázek 2.5: Blokové schéma Output režimu [17]

Příslušný signál je možno vytvořit pomocí softwaru nebo za pomoci obvodu s komparátorem a časovači. Pokud bychom chtěli generovat signál pomocí

softwaru, mohli bychom to udělat například tak, že bychom po zavolání zpožďovací funkce přepnuli výstup daného pinu do logické "1"nebo do logické "0". Případně by se výstupní hodnota pinu dala přepínat pomocí přerušení a časovačů. Při zvolené mezní hodnotě časovače by došlo k vyvolání přerušení, ve kterém bychom změnili logickou úroveň. Hardwardová varianta PWM využivá časovače v PWM módu. Zvolenou hodnotou v Auto Reload registru nastavujeme základní periodu T_0, případně se základní perioda upraví předděličkou hodinového signálu. Registr ARR může čítat hodnoty vzestupně, sestupně a nebo symetricky čítat vzestupně a od mezní hodnoty sestupně. Pro nastavení požadované střídy nám slouží Capture Compare registr, jehož hodnota se porovnává s obsahem ARR a při shodné hodnotě dojde ke změně stavu pinu (v případě Low-True je výstupní pin ve stavu logické "1", dokud nedojde ke shodné hodnotě v registrech ARR a CCR). Výsledná frekvence je dána nasledující rovnicí.

$$T_{\text{PWM}} = (ARR + 1) * (\text{Prescaler} + 1) * T_{\text{Clock}} \tag{2.2}$$

V režimu Low-true je pak výsledná střída daná poměrem hodnot registrů CCR a ARR. Opačně je tomu v režimu High-true, kde je výstup v logické "0", pokud je hodnota registru CCR větší než hodnota v registru ARR. První rovnice udává střídu pro Low-true režim, druhá rovnice pak High-true. [17]

$$\text{Duty Cycle}_{(\text{Low}-\text{true})} = \frac{CCR}{ARR + 1} \tag{2.3}$$

$$\text{Duty Cycle}_{(\text{High}-\text{true})} = 1 - \frac{CCR}{ARR + 1} \tag{2.4}$$

2.2.2 Vnímání světla a lidské oko

Lidské oko nevnímá přijaté světlo lineárně ale logaritmicky. Jas nám udává intenzitu osvětlení (cd/m^2) a světelnost to, jak lidské oko vnímá světlo. To jak vnímáme světlo popisují tyto rovnice podle CIE 1931, kde L je světelnost a Y je jas. [18]

$$\text{L}^* = 903.3 * \text{Y}, \; pokud \; \text{Y} \leq 0.008856 \tag{2.5}$$

$$\text{L}^* = 119 * \text{Y}^{1/3} - 16, \; pokud \; \text{Y} > 0.008856 \tag{2.6}$$

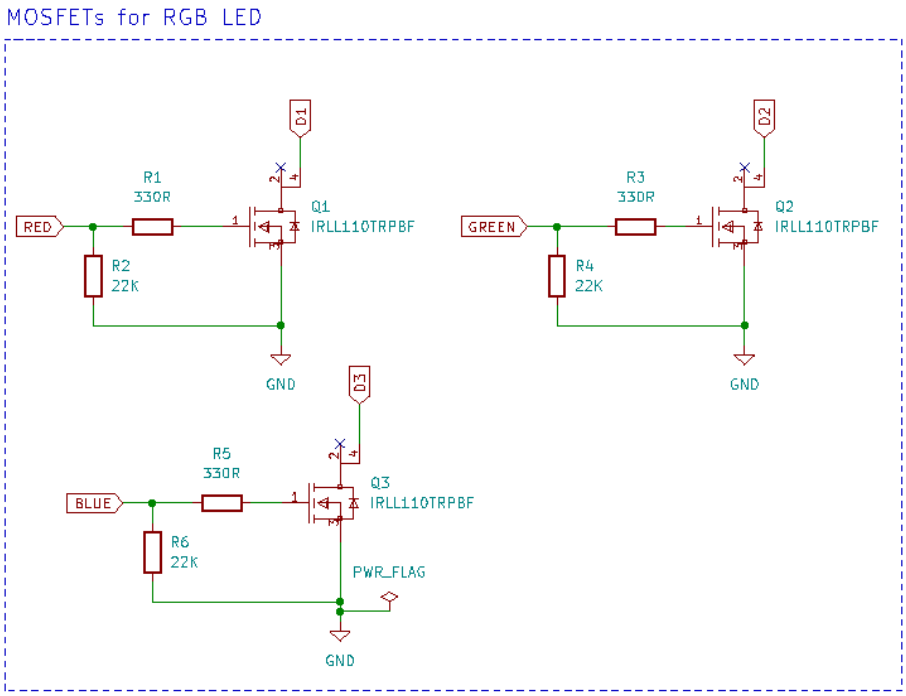
V aplikaci byl pak použit script pro přepočet PWM hodnot, tak aby bylo rozsvicování světla lépe znatelné. Script byl použit ze webových stránek od Jared Sansona. [18]

Kapitola 3

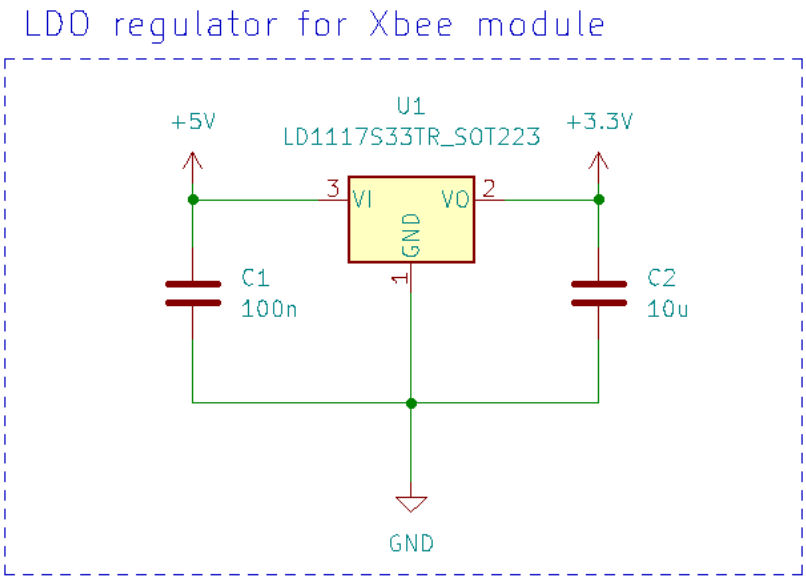
Driver pro RGB LED

3.1 Schéma zapojení

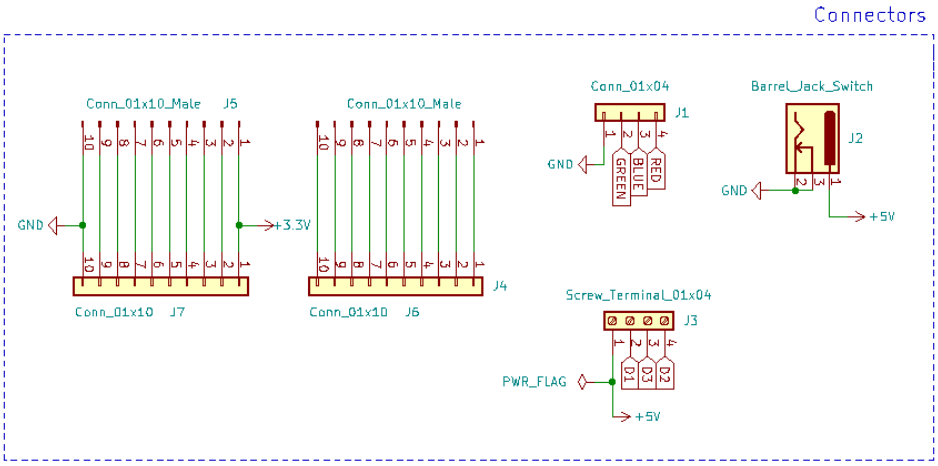
V našem případě je RGB LED řízena pomocí mikrokontroléru STM32G491RE. Daný mikrokontrolér používá 3.3V logiku, tedy pro stav logické nuly 0 V a pro stav logické jedničky 3.3 V. Světlo nemůžeme připojit napřímo k pinům mikrokontroléru, protože není schopný dodávat dostatečný proud a navíc bychom ho zničili, proto je nutné ovládat LED pomocí externího spínacího prvku, který se bude pomocí Nucleo desky pouze ovládat. Pro tento účel byl zvolen MOSFET tranzistor, který umožňuje spínat dostatečně velký proud diodou a jeho výhoda je v možnosti rychlého spínání. Jelikož se jedná o poměrně malé napětí byl zvolen MOSFET Vishay IRLL110, jehož prahové napětí je 2 V a už při 3 V je schopen spínat proud 1 A, což je pro naši aplikaci dostačující. Maximální proudy v jednotlivých diodách se pohybují ve stovkách mA (350 mA až 400 mA). Pro vytvoření vodivého kanálu mezi Drainem a Sourcem je potřeba nabít vstupní kapacitu mezi Gatem a Sourcem, nabíjecí proud kapacitoru je nutné omezit, aby nepřekročil maximální výstupní proud daného mikrokontroléru (pro Nucleo STM32G491RE je to hodnota 25 mA). Hodnota rezistoru pro Gate byla zvolena 330 Ω, tak aby se omezil nabíjecí proud na maximálně 10 mA. Dále je použit 22 kΩ rezistor k vybití kapacity mezi Gatem a Sourcem, aby se tranzistor vypnul i při odpojení řídícího napětí. K jednotlivým diodám (RGB) je předřazen 12 Ω resistor o výkon 2 W. Dále schéma obsahuje konektory pro připojení Nucleo kitu a pro napájení. Součastí schématu je LDO regulátor LD1117S33 od STMicroelectronics pro snížení napájecího napětí z 5 V na 3.3 V pro napájení desky Xbee, případně pro napájení samotného Nucleo kitu. Hodnoty kondenzátoru na vstupu a na výstupu byly zvoleny dle katalogového listu 100 μF a 10 μF. Výstup z regulátoru je připojen na pin 1 a 10.



Obrázek 3.1: Schéma - RGB LED driver



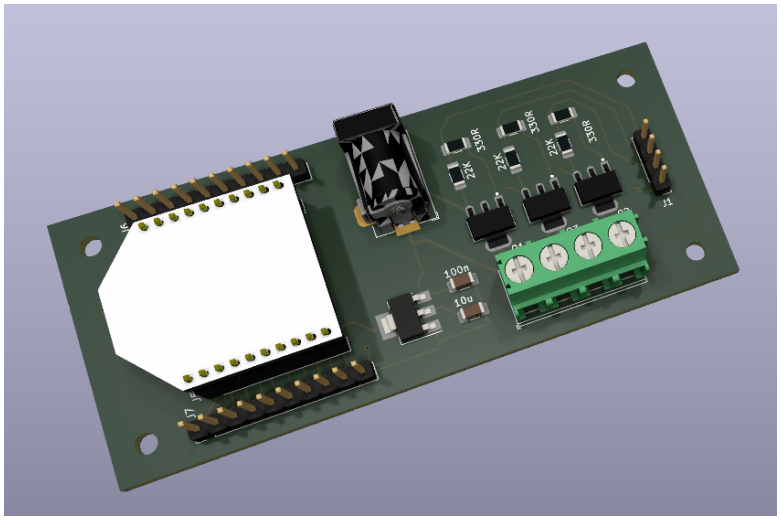
Obrázek 3.2: Schéma - LDO regulátor s výstupním napětím 3.3 V



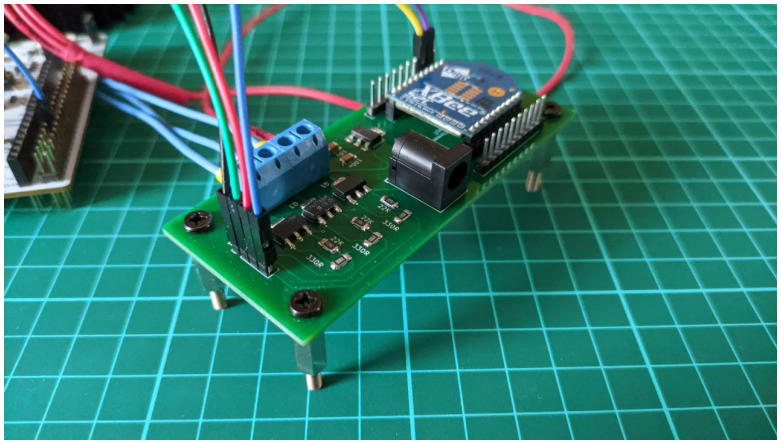
Obrázek 3.3: Schéma - konektory

3.2 Deska plošných spojů pro driver

Deska je dvouvrstvá o velikosti asi 90 x 40 mm s otvory v rozích pro montáž distančních sloupků. Deska byla navržena v programu KiCAD ve verzi 6.0, jedná se o open source a multiplatformní software pro návrh schémat a DPS. Šířky spojů jsou 0.3 mm pro signálové cesty a 0.5 mm pro napájení, dostačující pro naše proudové zatížení. Deska má tloušťku 1.6 mm a nechala se zhotovit ve společnosti JLCPCB. Níže je uvedený 3D model desky plošných spojů a pod ním vyrobená a osazená deska.



Obrázek 3.4: 3D model desky plošných spojů - RGB LED driver

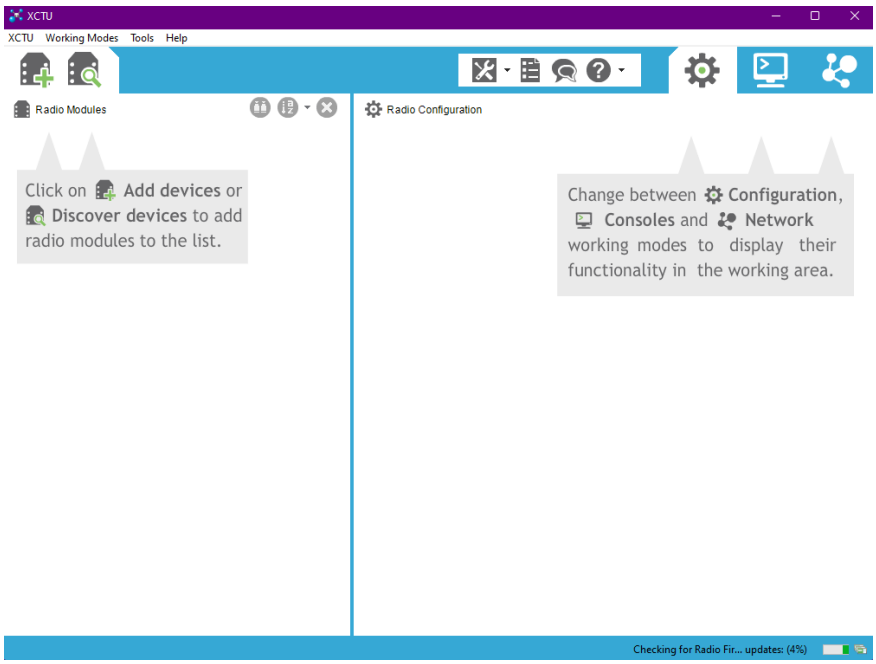


Obrázek 3.5: Zhotovená osazená deska - RGB LED driver

Kapitola 4

Návrh komunikačního protokolu pro dálkové ovládání

Xbee S2C moduly se konfigurují v softwaru XCTU. Pomocí něho můžeme nakonfigurovat MAC adresy zařízení, přiřadit síti její identifikátor, zvolit na jakém kanálu bude daná síť provozována. Dále program nabízí možnost vyhledat jednotlivé Xbee moduly nebo se dá stáhnout nový firmware do zařízení.



Obrázek 4.1: Rozhraní softwaru XCTU

4.1 Nastavení modulu Xbee

Rádio Xbee může pracovat ve dvou módech. Prvním je transparentní mód, ve kterém se data nijak neupravují a přímo se posílají do dalšího zařízení, to znamená, co vyšleme na jednom konci, tak bychom na druhém konci měli dostat ve stejné podobě. Jedná se o obdobu sériové linky s tím, že je přenos bezdrátový. Před samotnou komunikací v transparetním módu je nutné nastavit na zařízení MAC adresu cílové zařízení. Toto není problém v případě, že máme jen dva moduly, pro ty to postačuje nastavit pouze jednou. V případě více modulů je zapotřebí před každou zprávou vybrat cílové zařízení, tedy znovu nastavit cílovou MAC adresu. Adresu nastavíme pomocí XCTU, ale jestliže máme více zařízení, tak je to nepraktické. Proto existuje v transparentním modu konfigurační příkaz "+++" , pomocí tohoto textového řetězce se modul přepne do tohoho módu a za pomocí AT příkazů nastavíme cílovou adresu. Pokud mezi zařízeními nedochází ke komunikaci 10 s, modul se automaticky vrátí zpátky do transparentního módu.

Obrázek 4.2: Znázornění transparentního módu [19]

Druhým režimem je API (Application Programming Interface), ve kterém jsou data uspořádána do paketů. Takový paket obsahuje informace různé kategorie pro posílání zpráv nebo pro konfiguraci. Dále je v něm obsažena adresa odesílatele i příjemce, což usnadňuje komunikaci v případě více zařízení bez nutnosti přepnutí do příkazového módu.

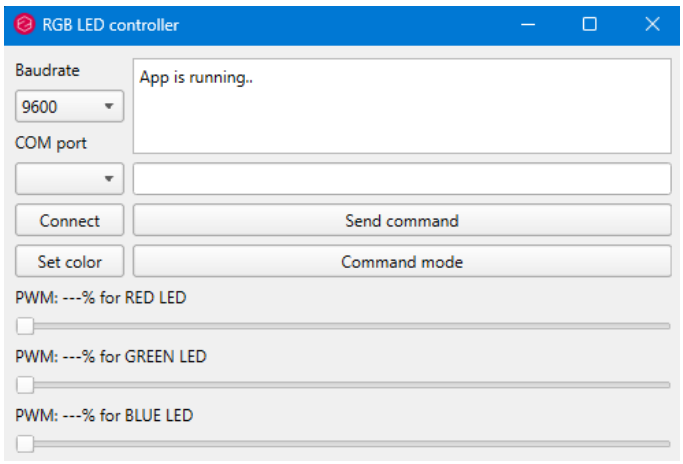
Obrázek 4.3: Znázornění API módu [20]

Jelikož jde v našem případě o komunikaci jen mezi řídicí deskou a Xbee modulem, vhodnější volba je transparentní mód, který nám poskytuje volnost při návrhu komunikačního protokolu. Samotná komunikace je navržena tak, že se přes transparentní mód posílá řetězec s hodnotami pro každý kanál (RGB). Samotná hodnota se zasílá jako řetězec, který je převrácený pro snažší zpracování na Nucleo desce. Jedná se o hodnotu Capture Compare registru, pomocí níž se mění střída PWM signálu. Řetězec vypadá například

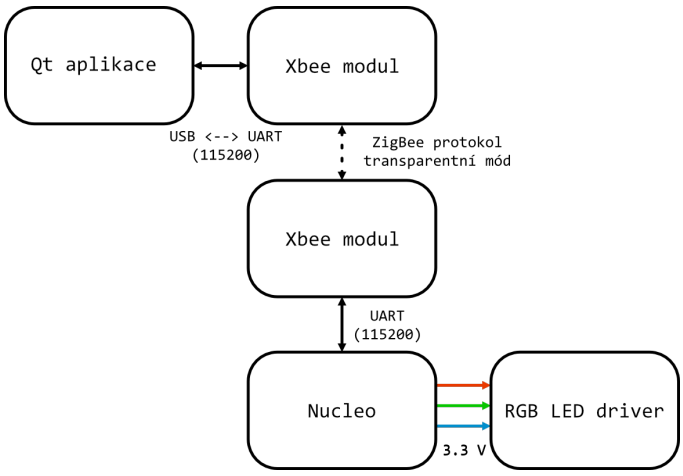
následovně "321R69G78B", tedy zaslali jsme hodnotu 123 pro R kanál, 96 pro G kanál a 87 pro B kanál. Dále si popíšeme pomocnou aplikaci v Pythonu.

4.2 Python Qt aplikace pro obsluhu RGB LED

Pro snadnou obsluhu byla napsána aplikace v Pythonu pomocí frameworku Qt. Ten umožňuje vytvářet grafické aplikace s tlačítky, grafy, ukazately apod.. V aplikaci si uživatel nejdříve nastaví Baudrate a COM port. Stiknutím tlačítka Connect se aplikace připojí k Xbee a může začít samotná komunikace. Pomocí posuvníku nastavujeme střídu pro jednotlivé kanály. Do textového pole nad tlačítkem Send Command se dají napsat příkazy pro ovládání LED. Tlačítko Command mode slouží pro přepnutí z transparentního módu do režimu příkazů ("+++"). Posuvníky se dají uzamknout pomocí tlačítka Set Color, díky tomu můžeme zvyšovat jas nastavené barvy.



Obrázek 4.4: Python Qt aplikace



Obrázek 4.5: Diagram propojení

Výše uvedený diagram popisuje, jak jednotlivé části mezi sebou komunikují. Oba Xbee moduly jsou nakonfigurovány jako koncová zařízení (transparentní mód). Popišme si nyní kód aplikace. V hlavní části se pouze vytvoří instance tříd *QApplication* a *Dashboard*, aby se vykreslilo hlavní okno aplikace. Pomocná třída *Dashboard* dědí ze třídy *QMainWindow*, která je součástí Qt frameworku. V této třídě je nastavena šířka a výška zobrazení okna aplikace a jsou v ní definovány Widgety.

```
#Main code
app = QApplication([])
app.setStyle("Fusion")
window = DashBoard()
window.show()
app.exec()
```

Ukázka kódu 4.1: Hlavní část aplikace

Dále jsou ve třídě *Dashboard* Widgety rozmístěny do horizontálních a vertikálních kontejnerů - Layoutů. v Qt frameworku jsou tzv. signály a sloty, kde každý Widget má svoje předdefinované akce. To znamená, že například stisknutí tlačítka vyvolá signál a zavolá se daná připojená funkce.

```
class DashBoard(QMainWindow):
    def __init__(self):
        super().__init__()
        #Dashboard settings
        self.setWindowTitle("RGB LED controller")
        self.setGeometry(750,350, 500,300)
        self.setWindowIcon(QIcon('smalldiscordlogoweb.png'))

        #Serial
        self.serial_port = serial.Serial()

        #Widgets
        self.btn_connect = QPushButton("Connect")
        self.btn_command = QPushButton("Send command")
        self.btn_color = QPushButton("Set color")
        self.btn_mode = QPushButton("Command mode")
        self.combo_baudrate = QComboBox()
        self.combo_port = QComboBox()
        self.label_baudrate = QLabel("Baudrate")
        self.label_port = QLabel("COM port")
        self.label_red = QLabel("PWM: ---% for RED LED")
        self.label_green = QLabel("PWM: ---% for GREEN LED")
        self.label_blue = QLabel("PWM: ---% for BLUE LED")
        self.textbox = QPlainTextEdit()
        self.textbox.setPlainText("App is running....")
        self.commandline = QLineEdit()
        self.slider_r = QSlider(Qt.Orientation.Horizontal)
        self.slider_g = QSlider(Qt.Orientation.Horizontal)
        self.slider_b = QSlider(Qt.Orientation.Horizontal)
        self.slider_r.setMaximum(250)
        self.slider_g.setMaximum(250)
        self.slider_b.setMaximum(250)
```

Ukázka kódu 4.2: Třída Dashboard

```
def connect_port(self):
    if(self.serial_port.isOpen() == True):
        self.serial_port.close()
        self.textbox.setPlainText("Disonnected!")
        self.btn_connect.setText("Connect")
    else:
        try:
            self.serial_port.port = self.combo_port.current
            Text()
            self.serial_port.baudrate = int(self.combo_baudrate.
            currentText())
            self.serial_port.timeout = 0
            self.serial_port.open()
            self.textbox.setPlainText("Connected!")
            self.btn_connect.setText("Disconnect")

        except:
            self.textbox.setPlainText("Cannot open PORT!")

self.btn_connect.clicked.connect(self.connect_port)
self.btn_command.clicked.connect(self.send_command)
self.btn_mode.clicked.connect(self.command_mode)
self.btn_color.clicked.connect(self.set_color)
```

Ukázka kódu 4.3: Funkce connect_port a její napojení na signál stisknutí tlačítka

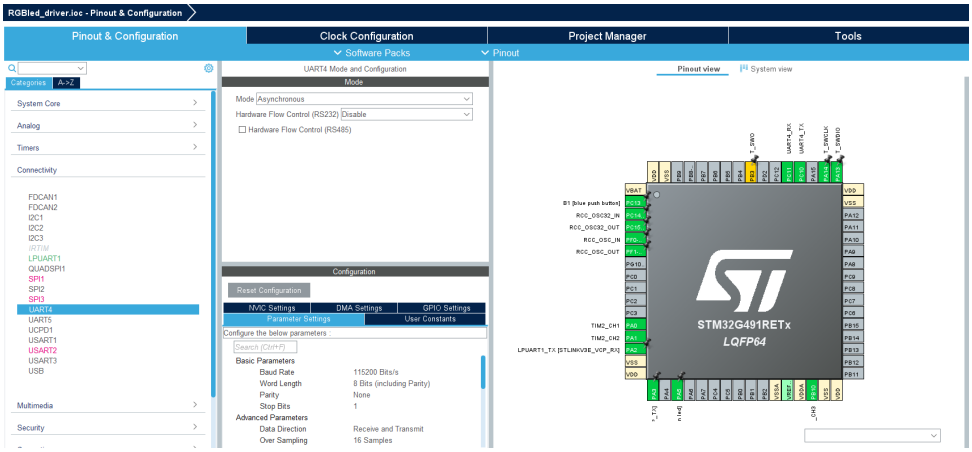
```
def PWM_g(self):
    if(self.serial_port.isOpen() and self.locker == 0):
        for i in str(self.L[self.slider_g.value()])[::-1]+":G":
            self.serial_port.write(i.encode())
            time.sleep(0.00001)
        self.label_green.setText("PWM: "+ str(round(self.
        slider_g.value()/250*100,2))+ " % - GREEN LED")
```

Ukázka kódu 4.4: Funkce pro změnu střídy kanálu B posuvníkem (modrá barva)

Třída definuje ještě další funkce pro obsluhu posuvníků a zasílání dat do Xbee modulu. Aplikace komunikuje s modulem pomocí knihovny pySerial.

4.3 Zpracování přijatého příkazu na Nucleo kitu

Program pro Nucleo kit STM32G491RE byl napsán v prostředí STMCube IDE a v jazyce C s použitím HAL (Hardware Abstract Layer) knihoven. Na obrázku níže je okno s nastavením periferií, pinů, registrů a zdroje hodinového signálu. Komunikace mezi Xbee a Nucleo deskou probíhá pomocí periferie UART4 při rychlosti 115200 Baudrate, kde byla zvýšena vstupní frekvence zdroje hodin, aby bylo možné komunikaci na této rychlosti provozovat. Pro příjem znaků se využívá přerušení pomocí DMA (Direct Memory Access), které zaručí rychlý přenos dat. Zjednodušeně se jedná o periferii. která přesune data z přijímacího datového registru UART do uživatelské paměti bez přítomnosti procesoru. Přerušení je nutné povolit v záložce pro periferii UART4. Ke generování PWM signálu byl využit 32-bitový TIMER2, který má 4 kanály. Hodnoty ARR registru a předděličky byly nastavené tak, aby výsledná frekvence PWM signálu byla 1 kHz. Je to dostatečná hodnota na to, aby lidské oko nevnímalo blikání RGB LED světla.



Obrázek 4.6: Vývojové prostředí STM32CubeIDE

Funkce *HAL_UART_RxCpltCallback* má atribut _weak, to znamená, že se ji můžeme sami předefinovat. Níže je naše definovaná funkce, která se zavolá při přijmu jednoho bytu. Ve funkci se kontroluje zdali je přijatý byte číslo a nebo znak "+". Po příjmu čísla a znaku barevného kanálu dojde k přiřazení hodnoty do příslušného CCR registru pro nastavení střídy.

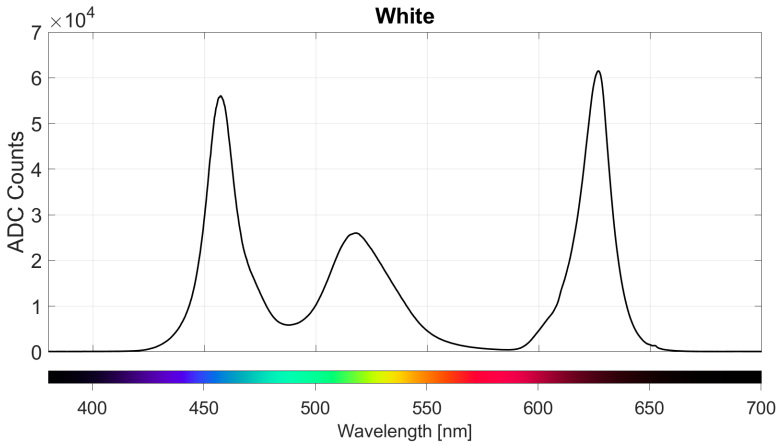
```
//Interrupt code
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
  //HAL_UART_Transmit(&huart4, rx_message, 1, 10);
  if((char)rx_message[0]-'0'>=0 && (char)rx_message[0]-'0'<=9)
  {
    CCR_value += ((char)rx_message[0] - '0')*temp;
    temp*= 10;
  }
  else
  {
    if((char)rx_message[0] == 'R')
    {
      TIM2->CCR1 = CCR_value;
      CCR_value = 0;
      temp = 1;
    }
    else if((char)rx_message[0] == 'G')
    {
      TIM2->CCR2 = CCR_value;
      CCR_value = 0;
      temp = 1;
    }
    else if((char)rx_message[0] == 'B')
    {
      TIM2->CCR3 = CCR_value;
      CCR_value = 0;
      temp = 1;
    }
    else if((char)rx_message[0] == '+')
        continue;
  }
  HAL_UART_Receive_DMA(&huart4, rx_message, 1);
}
```

Ukázka kódu 4.5: Definice funkce přerušení při přijmu jednoho bytu

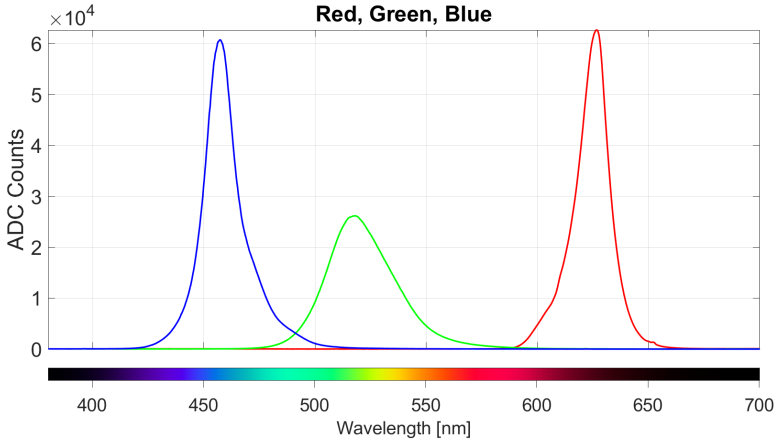
Kapitola 5

Měření spektra RGB LED

Pomocí spektrofotometru se naměřila jednotlivá spektra nejdříve pro každý jednotlivý RGB kanál a poté pro zvolené barvy. Měřicí rozsah spektrofotometru je od 176 nm do 1100 nm. Při měření byl nastaven integrační čas 0.3 s s průměrováním 100. Změřených dat bylo pro každou barvu asi 2800. Naměřená data se naimportovala do Matlab prostředí pro vykreslení grafů, kde byla využita data pouze od 380 nm do 780 nm vlnové délky. K vykreslení grafů se použila funkce *plot* a *spectrumColor* z toolboxu DIPUM3E. Jedná se o dodatečnou knihovnu do Matlab prostředí ke knize Digital Image Processing Using MATLAB, Rafel C. Gonzalez. Bílá barva byla vytvořena při 100% střídě všech tří barevných kanálů. Červená, zelená a modrá barva je také měřena při 100% střídě signálu.

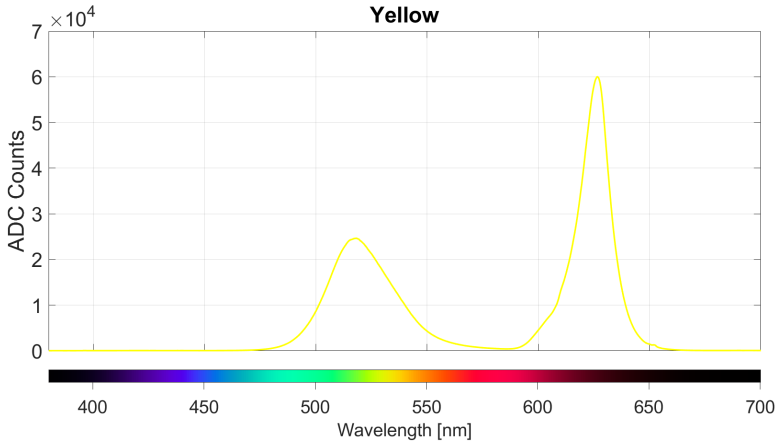


Obrázek 5.1: Bílá barva - spektrum

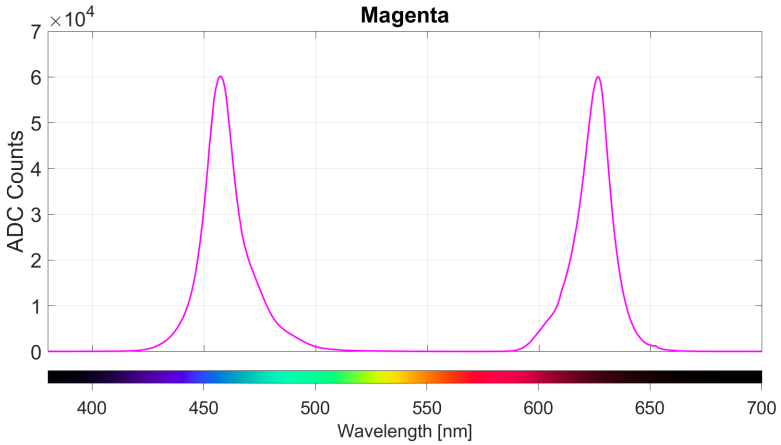


Obrázek 5.2: Červená, zelená a modrá barva - spektrum

Dále se nastavila žlutá a fialová barva světla. Jednotlivá spektra nastavených barev jsou níže.



Obrázek 5.3: Žlutá - spektrum

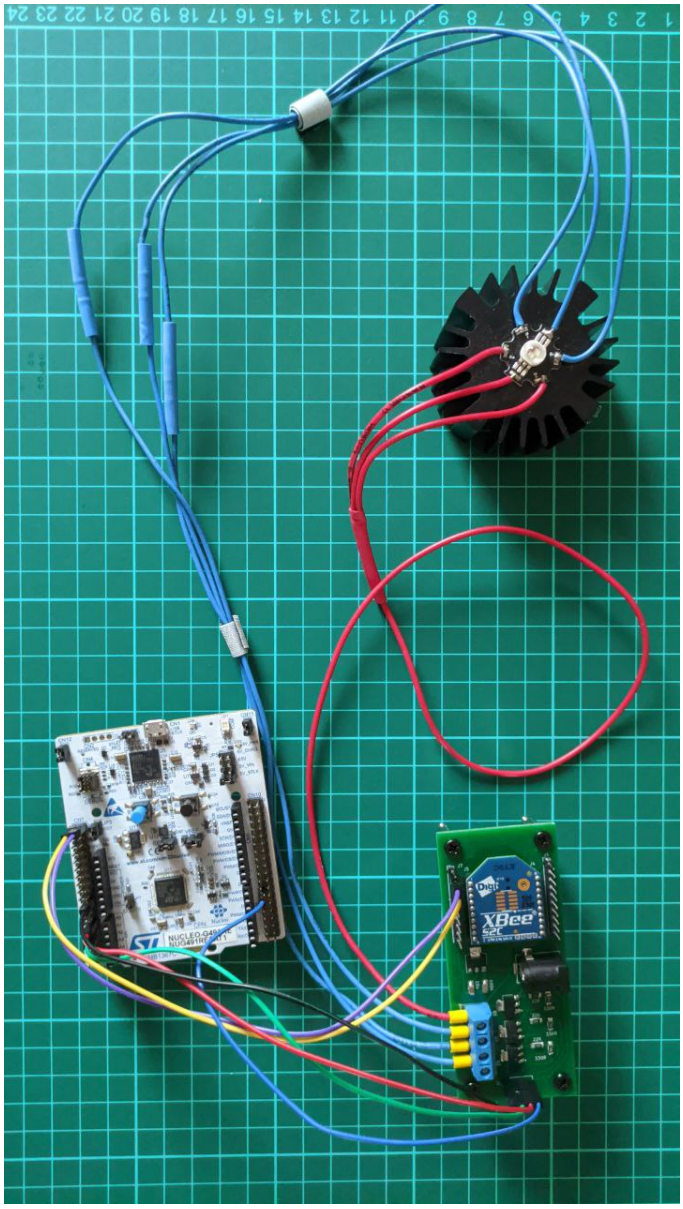


Obrázek 5.4: Fialová - spektrum

Kapitola 6

Závěr

Cílem této práce bylo navrhnout řídicí obvod pro RGB LED a komunikační protokol pro její dalkové ovládání. Ke komunikaci byl použit protokol ZigBee, který obstarávaly 2 Xbee moduly. Vytvořená síť je typu point-to-point. Řízení jednotlivých barevných kanálů LED je pomocí logických MOSFET tranzistorů, pro které je 3.3 V logika dostatečná. Pro tento účel byla navržena dvouvrstvá deska plošných spojů zhotovená v čínské společnosti JCLPCB. MOSFET tranzistory jsou řízeny pomocí Nucleo kitu, ke kterému je připojen Xbee modul komunikující s druhým modulem v transparetním módu. Dále byla vytvořena grafická Python aplikace umožňující nezávislé řízení jednotlivých barevných kanálů RGB. Pomocí spektrometru jsme změřili vlastnosti RGB LED nejdříve pro jednotlivé RGB kanály a pak pro několik zvolených barev. Případné vylepšení by spočívalo v možnosti měnit frekvenci PWM signálu v Python aplikaci tak, aby se vždy nastavila vhodná frekvence například při snímání světla kamerou. Dále by se mohlo použít více světel v mesh síti a vytvořit pomocí nich světelné efekty. Také by bylo možné RGB světlo připojit k nějakému chytrému asistentovi jako například Google Assistent nebo Amazon Alexa.



Obrázek 6.1: Finální navržené zařízení

Literatura

[1] SERPANOS, Dimitrios a Marilyn WOLF, 2018. Internet-of-Things (IoT) Systems: Architectures, Algorithms, Methodologies. Imprint: Springer, 102 s. ISBN 978-3-319-69715-4. Dostupné také z: https://link.springer.com/book/10.1007/978-3-319-69715-4

[2] Bluetooth logo [online]. In: . [cit. 2022-03-30]. Dostupné z: https://en.wikipedia.org/wiki/File:Bluetooth-logo.svg

[3] AFANEH, Mohammad, 2018. Intro to Bluetooth Low Energy: The easiest way to learn BLE [online]. [cit. 2022-04-25]. ISBN 1790198151. Dostupné z: https://www.novelbits.io/introduction-to-bluetooth-low-energy-book/

[4] Logo Zigbee [online]. In: . [cit. 2022-03-30]. Dostupné z: https://www.smartamenities.cz/rubriky/technologie/

[5] FALUDI, Robert, 2010. Building wireless sensor networks. Beijing: O'Reilly, 321 s. První. ISBN 9780596807733. Dostupné také z: https://www.oreilly.com/library/view/building-wireless-sensor/780596807757/

[6] Thread logo [online]. In: . [cit. 2022-03-31]. Dostupné z: https://rowoo.co.uk/wp/aiot/iot-technology/iot-protocols/thread/

[7] Thread adressing. In: Openthread [online]. [cit. 2022-04-26]. Dostupné z: https://openthread.io/guides/thread-primer/ipv6-addressing

[8] LoRaWAN logo. In: Lupa [online]. [cit. 2022-04-26]. Dostupné z: https://www.lupa.cz/clanky/tri-moznosti-jak-si-zprovoznit-vlastni-sit-lorawan-pro-internet-veci/

[9] LoRa. TheThingsNetwork [online]. [cit. 2022-05-17]. Dostupné z: https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/

[10] P-NUCLEO-WB55. In: Reichelt.com [online]. [cit. 2022-03-23]. Dostupné z: https://www.reichelt.com/fi/en/nucleo-68-stm32wb55-development-kit-bluetooth-5-incl-usb-dong-p-nucleo-wb55-p303098.html?&nbc=1

[11] In: St.com [online]. [cit. 2022-03-23]. Dostupné z: https://www.st.com/en/microcontrollers-microprocessors/stm32-wireless-mcus.html

[12] Xbee S2C [online]. In: . [cit. 2022-03-23]. Dostupné z: https://cz.farnell.com/digi-international/xkb2-a2t-wwc/dev-kit-xbee-s2c-ieee-802-15-4/dp/2766694

[13] IoT Clouds. In: Educba [online]. [cit. 2022-04-28]. Dostupné z: https://www.educba.com/iot-cloud-platforms/

[14] In: Wikipedia [online]. [cit. 2022-03-15]. Dostupné z: https://cs.wikipedia.org/wiki/MOSFET

[15] SPSMOH [online]. [cit. 2022-05-17]. Dostupné z: http://old.spsemoh.cz/vyuka/zel/tranzistory-unip.htm

[16] Mosfet charakteristika. In: Electronics-tutorials [online]. [cit. 2022-04-28]. Dostupné z: https://www.electronics-tutorials.ws/transistor

[17] Timers [online]. In: . [cit. 2022-04-28]. Dostupné z: https://www.ele.uri.edu/ qyang/ele547/Lecture4Timer2019.pdf

[18] Jared Geek [online]. [cit. 2022-05-02]. Dostupné z: https://jared.geek.nz/2013/feb/linear-led-pwm

[19] Transparent mode. In: Digi [online]. [cit. 2022-04-28]. Dostupné z:https://www.digi.com/resources/documentation/Digidocs /90001456-13/Default.htm#concepts/c_how_xbees_communicate.htm?TocPath =How%2520XBee%2520devices%2520work%257C_____1

[20] API mode. In: Digi [online]. [cit. 2022-04-28]. Dostupné z: https://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#concepts/c_api_mode_detailed.htm?TocPath =XBee%2520API%2520mode%257C_____1

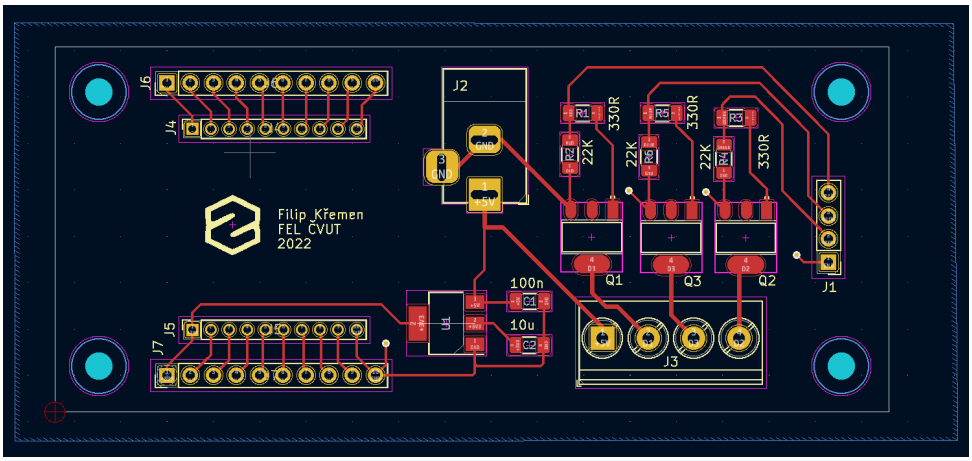
Příloha A

Zdrojové kódy pro Nucleo a Python Qt aplikaci

Dostupné z: https://gitlab.fel.cvut.cz/kremefil/rgb-led-svetlo-se-vzdalenym-ovladanim

Příloha B

Návrh DPS



Obrázek B.1: PCB layout

Dostupné z: https://gitlab.fel.cvut.cz/kremefil/rgb-led-svetlo-se-vzdalenym-ovladanim