

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 10: Elektrotechnika, elektronika a telekomunikace

SPBox

Adam Křen, Jiří Sankot
Plzeňský kraj

Plzeň, 2024

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 10: Elektrotechnika, elektronika a telekomunikace

SPBox

SBBox

Autoři: Adam Křen, Jiří Sankot

Škola: Vyšší odborná škola a Střední průmyslová škola
elektrotechnická Plzeň, Koterovská 828/85, 326 00 Plzeň

Kraj: Plzeňský kraj

Konzultant: Josef Fořt

Plzeň, 2024

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Plzni dne 13. 3. 2024 Jiří Sankot Adam Křen

Poděkování

Chtěli bychom poděkovat za bezednou pomoc kolegům a obětavým učitelům Josefu Fořtovi a Janu Drexlerovi, kteří se nám vždy snaží pomoci svými radami. Mnohokrát by se bez jejich rad tento projekt zasekl na jednom místě. Dále bychom chtěli poděkovat firmě TData za hmotnou podporu při vývoji a poskytnutí vybavení potřebného pro dokončení projektu.

Anotace

Tato práce se zabývá vytvořením zařízení, které pomáhá uživateli v užívání léků. Uživatel si nastaví přes webové rozhraní, kolik prášků denně bude užívat a časy léků. Zařízení monitoruje léky v políčkách a upozorňuje na užití léku světelnou signalizací. Pokud uživatel prášek nevybere rozezná se zvukové upozornění. Stav prášků v políčku se zobrazuje na webovém portálu. Díky využití komunikační technologie NB-IoT, uživatel nenastavuje žádné informace o síti, jako při využití jiné sítě. Zařízení je možné používat všude, kde má Vodafone NB-IoT pokrytí.

Klíčová slova

Zdravotnictví, léky, kapacitní snímání

Annotation

The aim of this project is to develop a device that assists users in managing their medication. Users can set parameters (such as the number of pills per day and the times at which the pills are taken) via a web application. The device keeps track of the pills and uses light signals to remind the user when it's time to take a pill. If the user forgets, a sound alarm is triggered. The status of the pill is displayed on the web portal. The device utilizes NB-IoT communication technology, eliminating the need for the user to configure network settings. The device can be used anywhere there is Vodafone NB-IoT coverage.

Keywords

pills, health care, capacitive sensing

OBSAH

Obsah	5
Úvod.....	7
1 Předchozí verze projektu a změny v nové verzi	8
2 Vize řešení	9
3 Hardware.....	12
3.1 Blokové schéma zařízení	13
3.2 Použité vývojové prostředí pro návrh desky plošných spojů a modelování krabice 14	
3.2.1 AUTODESK FUSION 360	14
3.3 Klíčové komponenty	15
3.3.1 PIC32CM5164JH01064 Microchip	15
3.3.2 BC660K QUECTEL	20
3.3.3 CMI-9653S-SMT-TR CUI Devices.....	22
3.3.4 Kapacitní senzory	23
3.3.5 Akumulátor L584070 CELLEVIA BATTERIES.....	27
3.3.6 IN-PI42TASPRGPB Inolux.....	28
3.3.7 MCP73833T Microchip	31
3.4 STC3115 STMicroelectronics.....	33
3.4.1 Základní informace	33
3.4.2 TPS63021DSJT Texas Instruments	34
3.4.3 Napěťový stabilizátor TPS61240.....	36
3.4.4 USB C Global Connector Technology (GCT).....	37
3.4.5 Signalizační LED	38
4 Návrh PCB – VRSTVA 1	40
5 Návrh PCB – VRSTVA 2.....	41
6 Návrh PCB – VRSTVA 3	42
7 Návrh PCB – VRSTVA 4.....	43
8 výroba, průběh osazování a následně vzniklé chyby	44
9 Návrh krabice.....	46
10 Návrh loga projektu	50
11 Použité komponenty z pohledu softwaru.....	52

11.1	PIC32CM5164JH01064.....	52
11.2	FreeRTOS	53
11.3	Kapacitní snímání	55
11.3.1	Testování PTC periferie.....	56
11.4	BC660K.....	57
11.4.1	Komunikační technologie NB-IoT	57
11.4.2	Testování komunikace s modulem pomocí konzole.....	59
11.4.3	Testování komunikace s modulem pomocí MCU	60
11.5	Programovatelná RGB LED s integrovaným obvodem WS2815	62
11.6	Testování komunikace s LED.....	62
11.7	STC3115IQT	64
12	Návrh softwarového řešení	66
12.1	Blokové schéma softwarového řešení projektu	66
12.2	Firmware.....	67
12.2.1	Původní koncept	67
12.2.2	Finální návrh firmware	67
12.3	Řešení aplikace	70
12.3.1	Oracle cloud.....	70
12.3.2	Ubuntu a Docker	70
12.3.3	Portainer.....	71
12.3.4	MySQL	71
12.4	Webová aplikace.....	73
13	Závěr	76
14	Seznam termínů	77
15	Seznam obrázků.....	79
16	Citovaná literatura.....	81
17	Seznam příloh	84

ÚVOD

Dnešní uspěchaná doba klade na každého z nás vysoké nároky a neustále nás něco dostává pod tlak. Člověk často musí řešit několik věcí najednou. To má za následek, že nemáme čas na; činnosti, které jsou méně důležité, ale jsou potřebné. Mezi takové činnosti se řadí například vynášení koše, mytí nádobí nebo třeba úklid. Bohužel lidé často zapomínají i na o hodně důležitější věci. Výjimkou není ani ta nejdůležitější, kterou není nic jiného než naše zdraví. Každý z nás jistě zažil situaci, kdy musel pravidelně užívat léky a z vlastních zkušeností vím, že hlavně, když je člověk zaneprázdněn, na to zapomene. Lidé, kteří užívají léky nejčastěji, jsou senioři. U nich na pravidelném užívání léků dokonce často závisí život. Jenže někteří z nich mají problém s pamětí.

Z těchto důvodů jsme se rozhodli, že zkusíme navrhnout a zprototypovat zařízení, které by dokázalo těmto problémům čelit. Toto zařízení má za úkol připomenout osobě, která zařízení používá, aby si vzal své léky včas a pokud možno i ve správném množství. Zařízení bylo původně zamýšleno hlavně pro starší lidi, kde by si našlo místo nejen u nich doma, ale také se může používat například v domově seniorů a v dalších lékařských a sociálních zařízeních. Samozřejmě jej můžou užívat jakkoli staří lidé, kteří si chtějí být jistý, že si své léky vezmou včas a ochrání tak své zdraví.

Zařízení dostalo vývojový název SPBox, což je ve zkratce znamená: smart pill box, v překladu chytrá krabička/box na prášky. Na první pohled jsou na krabičce vidět jednotlivá políčka na prášky, které jsou oddělené a jsou určeny právě pro prášky. Box na prášky není nijak velký, takže se vejde prakticky skoro kamkoliv.

Zařízení je vybaveno akumulátorem, aby nemuselo být stále na jednom místě a dalo se s ním snadno manipulovat. Je navrženo jako nízkoenergetická elektronika, často označována názvem low power. Oba mikrokontrolery využívají spánkové režimy. Když se akumulátor vybit, může ji jednoduše dobít pomocí USB-C kabelu, kterým se nabíjí například mobilní telefony. Díky barevné LED indikaci má uživatel přehled, kdy je potřeba box dobít. Podrobnější informace o stavu baterie uživatel nalezne v aplikaci.

Box snímá přítomnost léku pomocí kapacitního senzoru a když přijde čas na užití prášku, podsvítí dané políčko zeleně, při nevybrání začne políčko problikávat červenou barvou a rozezní se zvukové upozornění. Při vybrání prášku se světelné i zvukové upozornění vypne, aby dále uživatele nerušilo. Zařízení má v sobě modul, který se umí připojit na low power síť NB-IoT (zkratka pro označení „Narrowband IoT“), pomocí které zařízení komunikuje s aplikací. V ní si uživatel jednoduše zobrazí informace o stavu baterie, stav vybrání prášků a nastaví si čas, ve kterém si pravidelně prášek bere. Je možné také specifikovat počet prášků denně, informace o prášku a číslo na lékaře.

1 PŘEDCHOZÍ VERZE PROJEKTU A ZMĚNY V NOVÉ VERZI

Minulý rok jsme dělali na projektu, který plnil stejný účel jako ten letošní. Po zkušenostech s různými technologiemi z minulého roku jsme se rozhodli některé věci pozměnit a zachovat.

Princip fungování boxu zůstává stejný jako v roce minulém. Mikrokontroler pomocí senzorů zjišťuje přítomnost léku, kontroluje čas a když nastane čas na užití léku, uživatele upozorní. Nejdříve pomocí zeleného podsvícení a při nevybrání léku se políčko rozblíká červenou barvou a rozezní se zvukové upozornění. Box v časovém intervalu odesílá data do webové aplikace. Při nastavení parametrů uživatelem ve webové aplikaci se zařízení přenastaví.

Po zkušenostech a zpětné vazbě z obhajoby ročníkové práce jsme se rozhodli udělat v projektu několik klíčových změn. První hlavní změnou, uživatelem nejvíce rozeznatelnou, je rozšíření boxu ze 7 políček na 21. K tomuto kroku jsme přistoupili z důvodu doporučení porotou při obhajobě ročníkové práce. Zněla tak, že v praxi by zařízení se sedmi políčky nebylo moc využitelné, protože lidé často potřebují užívat i několik léků denně.

Další změnou je samotný princip detekování prášků. Minulý rok byl k detekování léku využíván optický senzor přiblížení VCNL36825T, který funguje na principu odrazu světla. Jelikož se jednalo o běžnou technologii, tak jsme se letos rozhodli pro detekci pomocí kapacitních senzorů.

Poslední významnou změnou je použití jiné přenosové technologie. Minulý rok jsme používali komunikační technologii LoRa a LoRaWAN. Zvolili jsme ji pro její energeticky nenáročné fungování. Narazili jsme ale na problémy s pokrytím. Přemýšleli jsme jakou technologii namísto LoRa zvolit a došli jsme k závěru, že využijeme NB-IoT. Jedná se také o technologii s nízkou spotřebou a má velmi výrazné pokrytí nejen v České republice. Nevýhodou je, že její používání je zpoplatněno operátorem. Se změnou používané komunikační technologie jsme přešli na kontrolování času v samotném zařízení. Je synchronizován přes komunikační síť NB-IoT z aplikačního serveru, v minulé verzi byl čas hlídán v aplikaci a do zařízení byla odeslána pouze upozornění na jeho shodu.

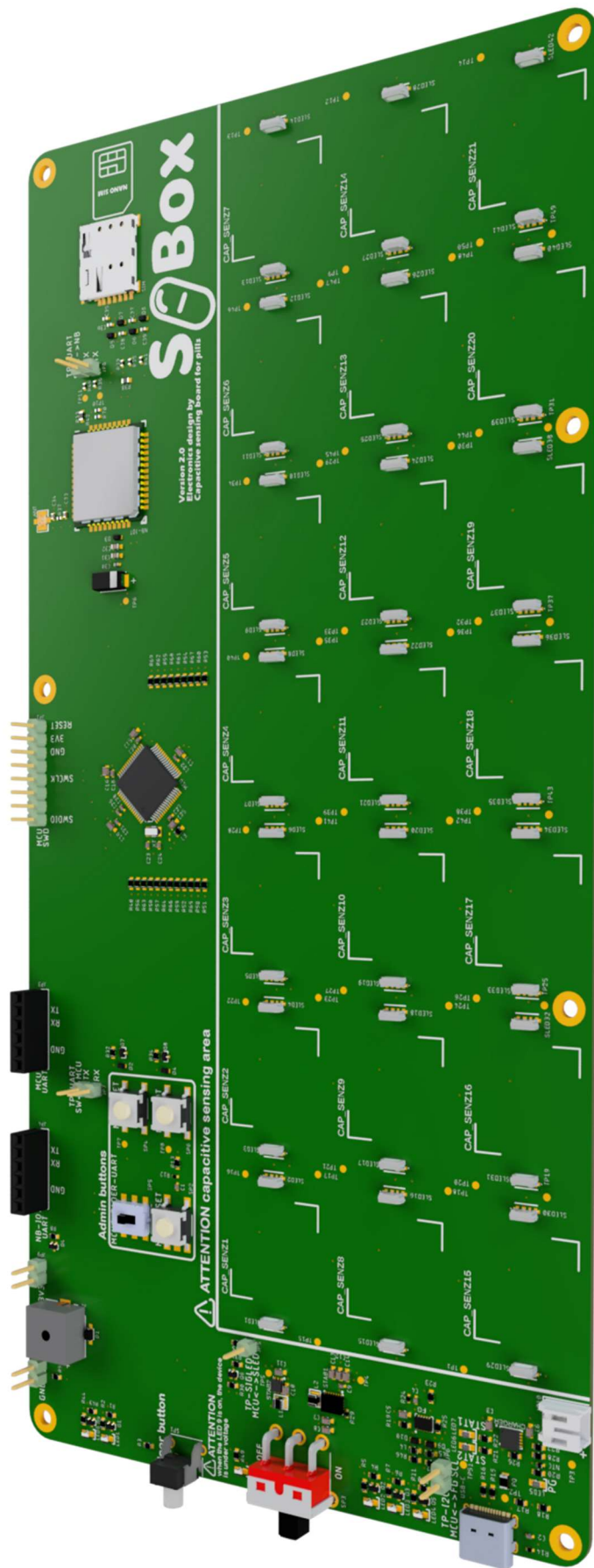
2 VIZE ŘEŠENÍ

Na trhu je mnoho podobných zařízení ať už se jedná o produkty mechanického typu, nebo elektrického. Klasické mechanické organizéry jsou jistota, ale uživatelé mimo přehlednost nepřidávají žádnou přidanou hodnotu. Správné užívání léku spočívá na uživateli, takže při jeho zapomenutí, nemá žádnou možnost připomenutí. Každopádně důvodem, proč si zakoupit tento jednoduchý organizér je cena, které elektrické chytré organizéry nejsou schopny dosáhnout. Dalším typem krabičky na léky, která je na trhu dostupná je elektronický organizér s nastavitelnými alarmy. Ty jsou nastavitelné pomocí uživatelských tlačítek a vestavěného displeje. Tento typ produktu je takový kompromis mezi naším zařízením a mechanickou verzí organizéru. U takového zařízení chybí detekce léků, a tak je nutné vyzvednutí léku potvrzovat pomocí uživatelského tlačítka. Nenalezneme u něj ani možnost vzdáleného ovládní, což může být pro některé uživatele nekomfortní. Výhodou takového typu produktu je cena, která je výrazně nižší než při použití bezdrátové komunikace a senzorů pro detekování léků. Některé produkty fungující bez snímání léků jsou doplněny o bezdrátovou komunikační technologii, která nahrazuje zabudované uživatelské rozhraní typu displeje a tlačítek. Nejčastějším typem komunikace je technologie Bluetooth, jež slouží pro komunikaci s lokálním zařízením. Ovládní organizéru je přesunuto do mobilní aplikace. Výhodou je bezplatná bezdrátová komunikace, oproti technologii NB-IoT. Poslední typ organizéru vyskytující již ovsahuje detekování léků. Ta je z velké části řešena pomocí světelných senzorů, tedy stejným způsobem jako v minulé verzi.

Po prohledání podobných produktů uvedených na trh a zjištění jejich kladných a záporných vlastností, jsme došli k rozhodnutí pokusit se některé z jejich slabších stránek vyřešit. Hlavním tématem byla technologie použitá k detekování léků. Po zkušenostech s optickými senzory fungujícími na principu světelného odrazu a nimi spojeným výrazným prodražením celkového designu jsme se rozhodli vyzkoušet zatím k těmto účelům konkurencí nepoužívanou technologii kapacitního snímání. Tato technologie by díky implementaci senzoru vytvořeného na desce plošných spojů snížila náklady s ním spojené.



Obrázek 1: 3D vizualizace kompletního boxu



Obrázek 2: 3D vizualizace desky plošných spojů

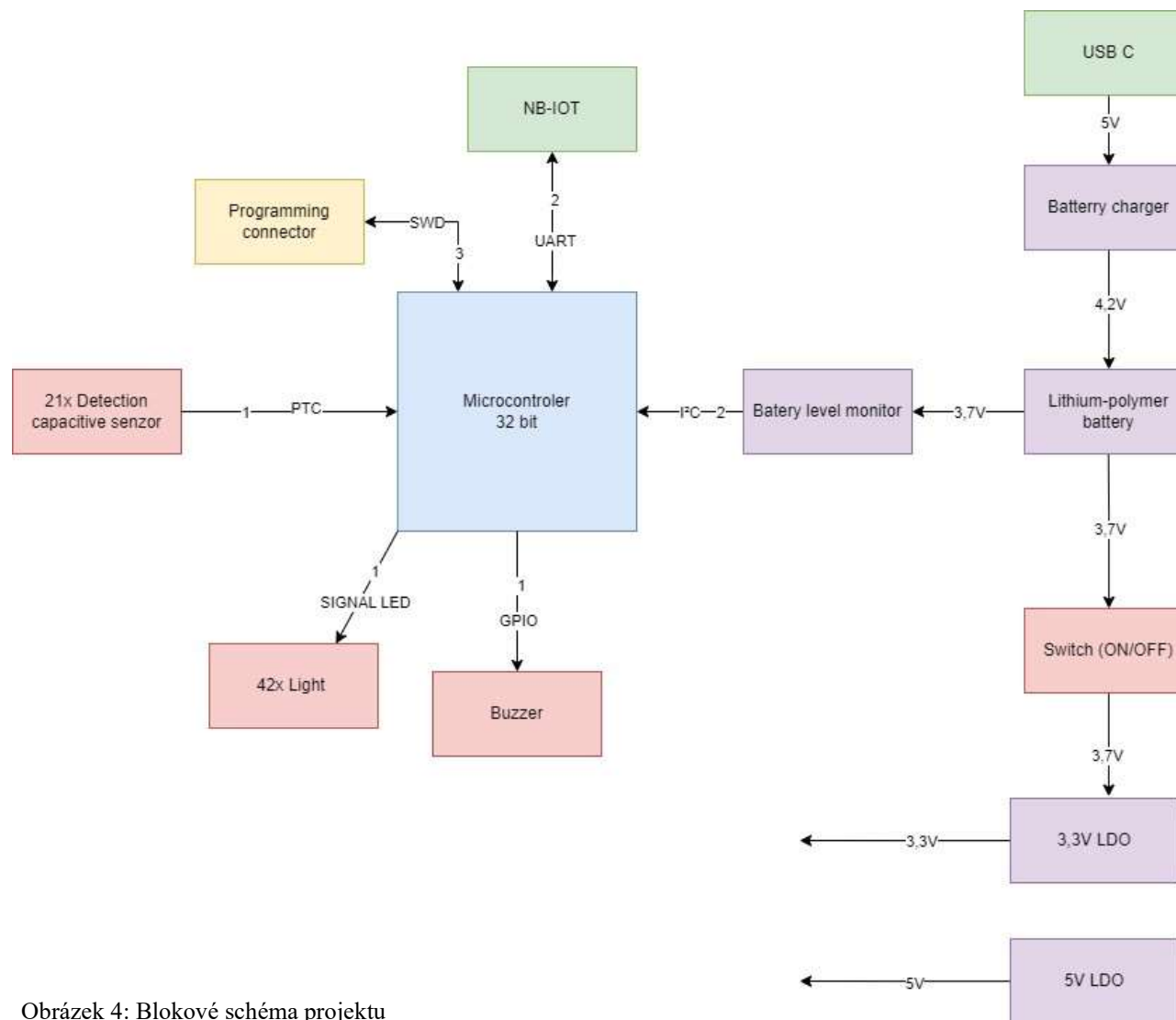
3 HARDWARE

Po fázi rozhodování a vymýšlení základních vlastností našeho výrobku a celkové koncepce toho, jak by zhruba zařízení mělo vypadat a fungovat, jsme se přesunuli do fáze hardwaru a začali jsme nápady promítat do blokových schémat a schémat zapojení.

Jako první se začal řešit problém s detekcí prášků v krabičce, aby když do krabičky prášky vložíme, krabička věděla, že v ní prášky jsou a mohla s nimi dále pracovat. Loni jsme použili pro detekci prášků takzvané senzory přiblížení, které fungovali na optické metodě. Nevýhoda tohoto řešení byla, že prášky se musely dostat do dráhy paprsku senzoru, aby je senzor zaregistroval. Docházelo zde k občasnému nedetekování prášků, přestože byl prášek v krabičce přítomen. Prášek se někdy přesunul na stranu boxíku a vychýlil se tak z detekční dráhy senzoru. Taky musely být v krabičce v každém boxíku otvory pro volný průchod paprsku, což bylo nežádoucí, protože mezi práškem a PCB byla pouze vzduchová mezera. Proto jsme letos zkusili jít jinou cestou a pokusili jsme se použít senzory kapacitní. Jelikož jsme většinou jako jádro, našich IOT zařízení používali mikrokontrolery od firmy Microchip, zjistili jsme, že jedna z mnoha funkcí, které mají, je periférie PTC (Peripheral Touch Controller), díky této periférii můžeme zpracovávat data o kapacitních senzorech a dále s nimi pracovat. Jedna z největších výhod tohoto řešení je možnost vlastního návrhu kapacitních senzorů a jejich přímá integrace v PCB. Kromě nemalého úsilí při jejich vývoji a testování nás nic nestojí, protože jejich cena je zahrnuta v ceně celého PCB a nemusí se tak platit za speciální senzory jako tomu bylo při řešení se senzory optickými. Obzvláště letos jsme oproti loňskému roku zvětšili kapacitu krabičky ze 7 na 21 boxíků, což v případě ceny loňského optického senzoru (kolem 50 Kč za kus), by rapidně zvýšilo celkové náklady na vývoj tohoto projektu.

Abychom mohli nějak signalizovat uživateli, z jakého boxíku by si měl vzít prášek, řešili jsme zde otázku použití LED. Jelikož je 21 jednotlivých boxíků, a na každý používáme dvě LED, které obklopují kapacitní senzor, aby boxík nebyl nasvícen nesymetricky jen z jedné strany, bylo by potřeba většího počtu pinů na našem mikrokontroleru. Proto jsme se rozhodli použít, speciální programovatelné LED diody, které mají v sobě chip a komunikují po jedné sběrnici což znamená, že na mikrokontroléru to zabere jen 1 pin. Tímto způsobem se ušetří za mikrokontrolér s větším počtem vývodů a celkově tím snížíme cenu projektu.

3.1 Blokové schéma zařízení



Obrázek 4: Blokové schéma projektu

3.2 Použité vývojové prostředí pro návrh desky plošných spojů a modelování krabice

3.2.1 AUTODESK FUSION 360

Fusion 360 je cloudový software pro 3D modelování, CAD, CAM, CAE a PCB určený k navrhování a výrobě produktů. Má kompletní výběr překladačů dat pro 50 různých typů souborů. Nabízí mnoho funkcí.

Nabízí přímé, povrchové, síťové a volné modelování, jakož i vytváření plechových dílů, soustružení a frézování a interaktivní sestavování. Přímé modelování je podporováno v režimech založených na historii i bez ní.

Nabízí skvělou funkci, která dovoluje vytvářet vlastní objekt na CNC stroji. Má nativní funkce pro počítačem podporovanou výrobu. Je možné provádět řezy pomocí bezplatných nástrojů postprocesoru softwaru nebo si vytvořit vlastní.

Pomocí funkcí pro návrh schémat, designu desek plošných spojů a možnosti routování můžete navrhnout svou vlastní desku plošných spojů. Fusion 360 dovoluje více lidem pracovat na stejném designu. Pro převod, změnu nebo exportování souborů není potřebný žádný přídavný software.¹

Program podporuje simulování různého typu namáhání a deformování modelu. Pomáhá odhalovat běžné poruchové stavy a úrovně napětí.



Obrázek 5: Logo aplikace Fusion 360, dostupné z: <https://in.pinterest.com/pin/268456827780030537/>

¹ (Autodesk, Inc., Nedatováno)

3.3 Klíčové komponenty

3.3.1 PIC32CM5164JH01064 | Microchip

3.3.1.1 Základní informace

V oblasti elektrotechniky se MCU, neboli Microcontroller Unit, či mikrokontroler, označuje za integrovaný obvod, který zahrnuje jádro procesoru, periferní zařízení (jako jsou GPIO piny, komunikační rozhraní, časovače) a paměť (RAM, ROM, Flash). Typicky obsahuje MCU jedno nebo více jader procesoru, které mohou být založeny na různých architekturách, jako je například ARM, AVR, PIC. Architektura MCU určuje mnoho vlastností, včetně rychlosti, spotřeby energie, dostupnosti periferních funkcí a také možností vývoje softwaru.² My používáme konkrétně mikrokontrolér od firmy Microchip a to model PIC32CM5164JH01064. Tento model je založený na architektuře Arm® Cortex®-M0+. Jeho napájecí napětí se pohybuje od 2,7 – 5,5 V. Má poměrně velkou paměť a to 512 KB Flash a 64 KB SRAM. Podporuje taky řadu různých periférií jako je CAN, CANFD, UART, SPI, I2C a RS-485³.

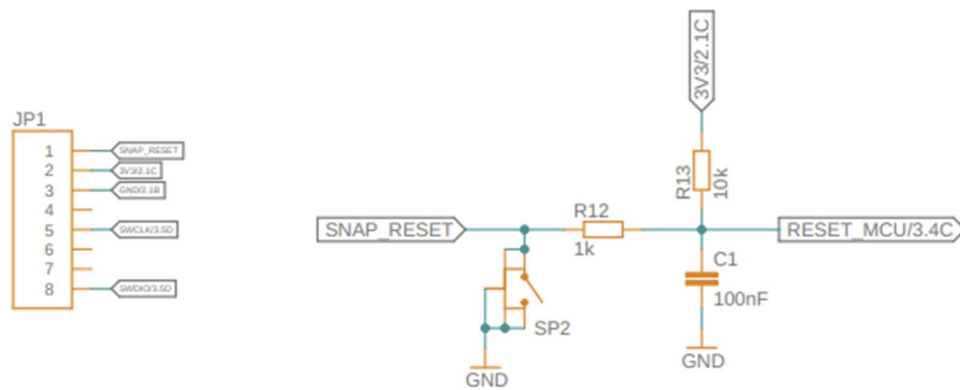
3.3.1.2 Použití v projektu

V našem projektu hlavně využíváme UART a I²C. Ale asi pro nás nejdůležitější periférií, kterou tento mikrokontroler má je vylepšený Peripheral Touch Controller (PTC), česky periferie pro ovládání kapacitních senzorů, která má až 256 (16 x 16) kanálů pro zapojení senzorů se vzájemnou kapacitou a až 32 kanálů pro připojení senzorů s vlastní kapacitou. Přičemž my máme 21 senzorů, pro možnost detekování 7 boxíků pro 7 dní x 3 boxíky 3 různé časové úseky, vybraní lékařský prášků. Nám se konkrétně nejlépe osvědčila metoda s vlastní kapacitou.

Pro programování mikrokontroleru používáme programovací piny RESET, SWDIO a SWDCLK jsou vyvedeny na osmi pinový header. MCU je možno resetovat i přes tlačítko, které je zapojeno tak, že při stisknutí tlačítka svede elektrický proud k zemi. Paralelně ke spínači je filtrační kondenzátor o kapacitě 100 nF. Přes pull-up rezistor je připojen k napájení, aby nedošlo k nechtěnému resetování MCU. Také je zde pro detekci zapojení USB do MCU přes odporový dělič, složený ze dvou 10 kΩ rezistorů, přivedeno napájecí napětí z USB.

² (Wikipedia otevřená encyklopedie, 2022)

³ (Microchip Technology Inc., Nedatováno)



Obrázek 6: Schéma zapojení headeru pro externí debugger a reset tlačítka

Mikrokontroler je také masterem sběrnice I²C, na které je připojen fuel gauge. Sběrnice pracuje ve Standart modu. I²C je na MCU připojeno na piny PA12 a PA13. Oba vodiče sběrnice mají 1 kΩ pull-up rezistory. Pull-up rezistory mají několik důležitých funkcí v I²C sběrnici, například zajištění logické úrovně HIGH. V I²C sběrnici jsou linky SDA (Serial Data Line) a SCL (Serial Clock Line) aktivně řízené zápornými impulzy (open-drain). Pull-up rezistory jsou připojeny mezi tyto linky a napájecí napětí, aby zajistily, že ve stavu HIGH jsou tyto linky udržovány na logické úrovni odpovídající napájecímu napětí. Také slouží jako filtr na odstranění šumu a je k ním také umístěna přepětěová ochrana, která je tvořena dvěma TVS diodami na 3,6 V.

Výpočet již zmíněných pull-up rezistorů byl proveden podle rovnic (1) a (2).

$$R_{P(MAX)} = \frac{t_r}{0,8473 \times C_b} = \frac{300 \times 10^{-9}}{0,8473 \times 200 \times 10^{-12}} = 1,77 \text{ k}\Omega$$

Rovnice 1: Výpočet maximální hodnoty pull-up rezistorů sběrnice

$R_{P(MAX)}$ - maximální hodnota odporu pull-up rezistorů

t_r - čas náběžné hrany signálu (pro fast mode je to maximálně 300 ns)

C_b - maximální kapacita sběrnice

$$R_{P(MIN)} = \frac{V_{CC} - V_{OL(MAX)}}{I_{OL}} = \frac{3,3 - 0,4}{3 \times 10^{-3}} = 967 \Omega$$

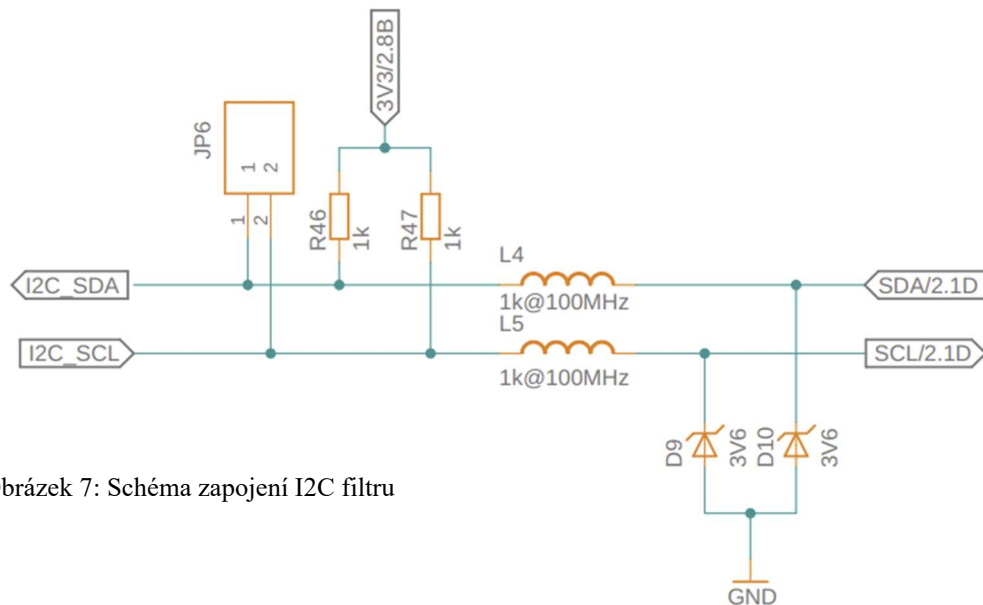
Rovnice 2:: Výpočet minimální hodnoty pull – up rezistorů sběrnice

V_{CC} je napětí přivedené na sběrnici (my budeme používat napětí 3,3 V)

$V_{OL(MAX)}$ - maximální napětí při stavu LOW – stavu low při spotřebě proudu 3 mA a $V_{CC} > 2 \text{ V}$ (pro fast mode 0,4 V)

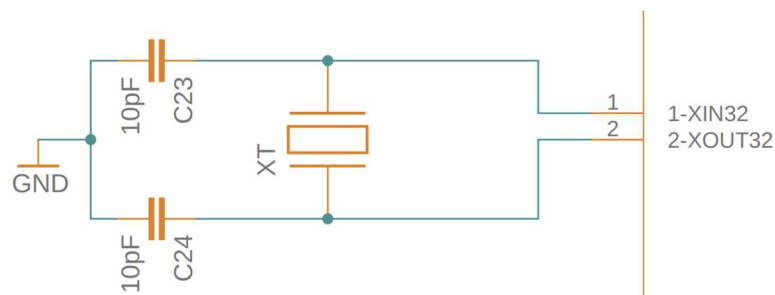
I_{OL} – maximální proud při stavu LOW pro ($V_{CC} > 2 \text{ V}$ je 3 mA).

Z vypočítaných odporů $R_{P(MAX)}$ a $R_{P(MIN)}$ byla zvolena hodnota pull-up rezistoru $1k\Omega$ ⁴



Obrázek 7: Schéma zapojení I2C filtru

V mikrokontrolerech se externí krystal často používá jako zdroj hodinového signálu pro synchronizaci operací v mikrokontroleru. Externí krystal poskytuje stabilní a přesný časový základ. V našem projektu používáme k mikrokontroleru také externí krystal. Níže je vidět dopočítání kondenzátorů v obvodu, aby obvod s externím krystalem fungoval správně. Kondenzátory pomáhají stabilizovat oscilaci krystalu tím, že poskytují stabilitu a vyrovnávají jeho impedanci. To je důležité pro udržení přesnosti a stability frekvence oscilace.



Obrázek 8: Schéma zapojení externího krystalu

⁴ (Texas Instruments Incorporated, 2015)

Parametry námi použitého krystalu:

Přesnost 20 ppm což je 0,002

$f = 32,768 \text{ kHz}$

$C_L = 9 \text{ pF}$

Parametry našeho mikrokontroleru:

$C_{XTAL_EFF} = 5,5 \text{ pF}$

$C_{PCB} = 1,5 \text{ pF na } 12,5 \text{ mm}$

Výpočet:

$$C1 = C2 = (2 \times C_L) - C_{XTAL_EFF} - (2 \times C_{PCB})$$

$$C1 = C2 = (2 \times 9) - 5,5 - (2 \times 0,75)$$

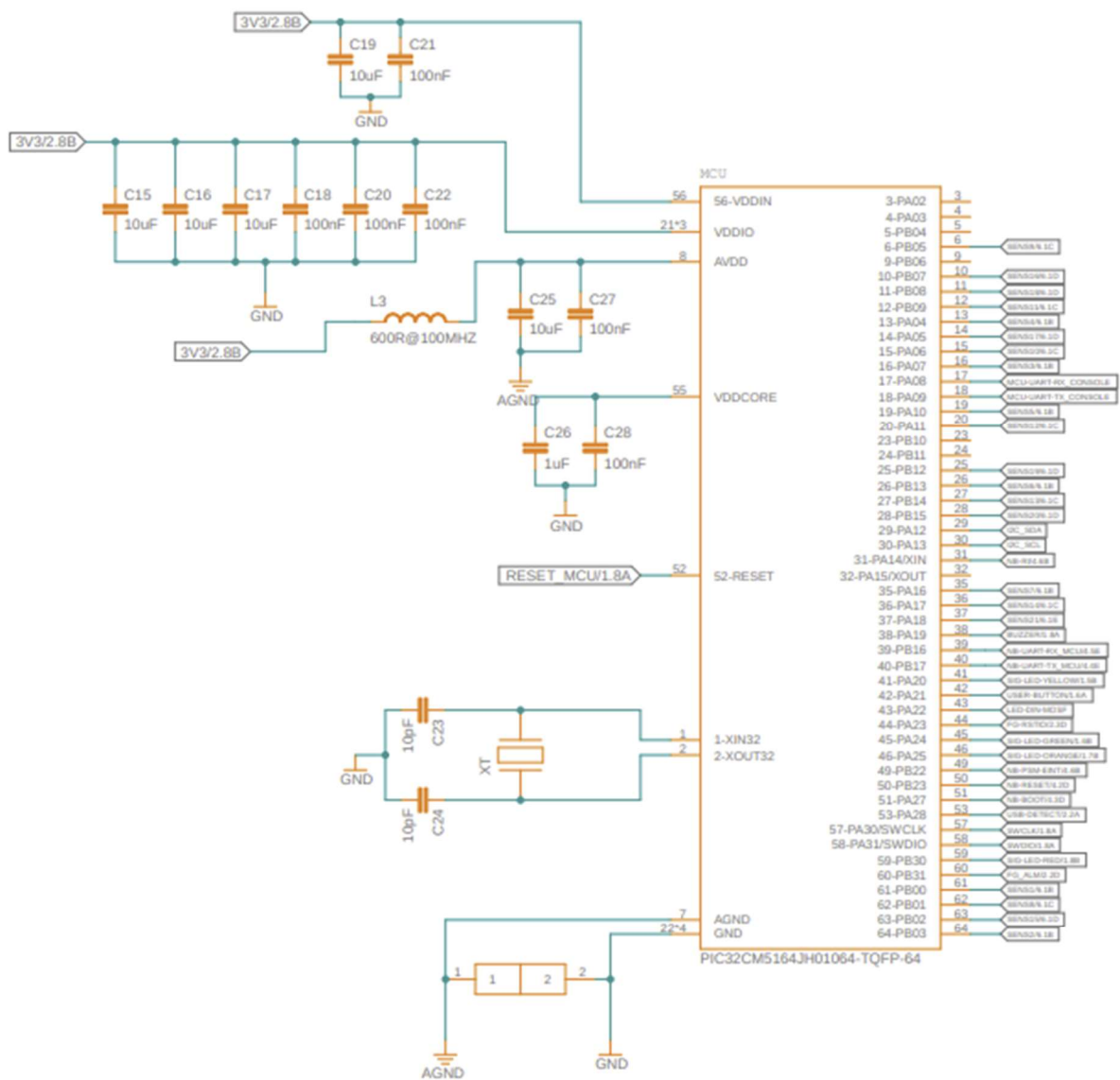
$$C1 = C2 = 11 \text{ pF}^5$$

Rovnice 3: Výpočet hodnot kondenzátorů v obvodu s externím krystalem



Obrázek 9: Mikrokontroler PIC32CM5164JH01064 v 64 pinovém pouzdře TQFP, dostupnéz:<https://www.microchip.com/en-us/product/pic32cm5164jh01064>

⁵ (Microchip Technology Inc., 2022)



Obrázek 10: Schéma zapojení MCU

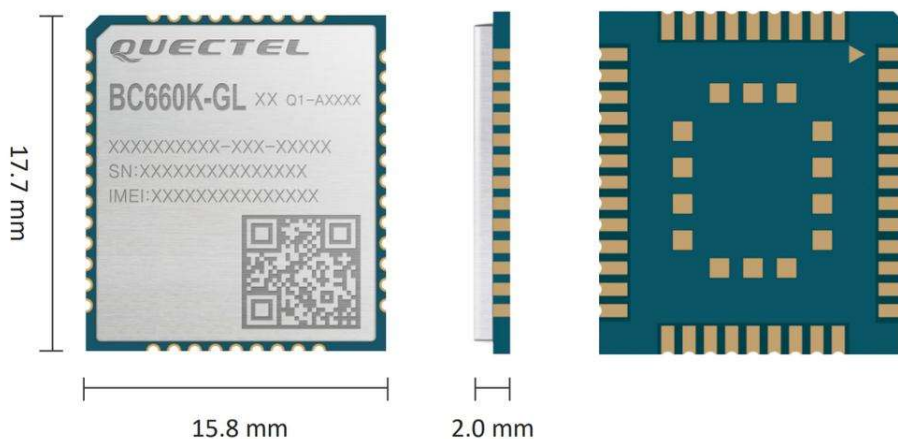
3.3.2 BC660K | QUECTEL

3.3.2.1 Základní informace

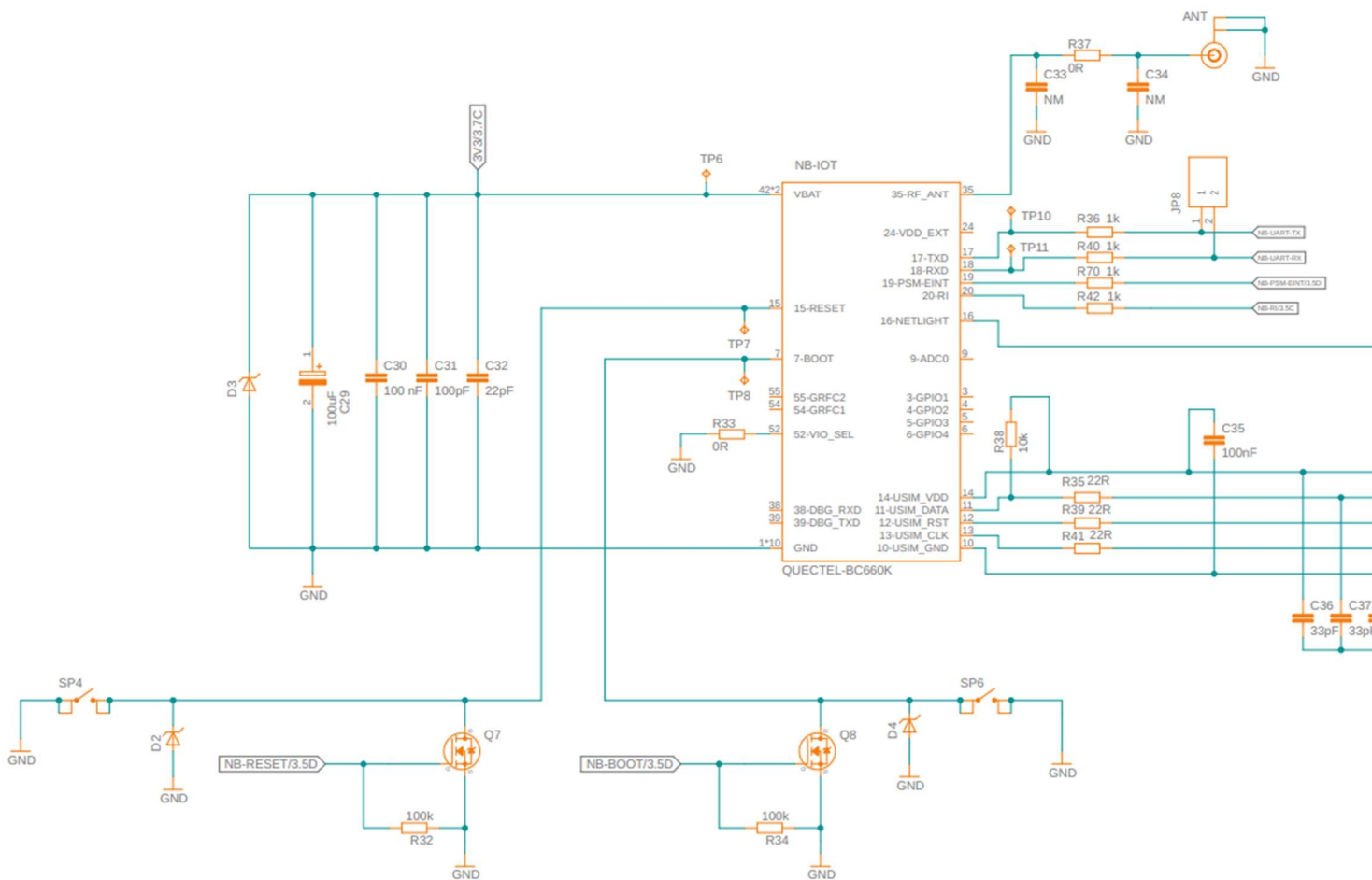
BC660K-GLAA-I03-SNASA je LTE Cat NB2 modul vyvinutý společností Quectel pro připojení k NB-IoT síti. Tento modul podporuje širokou škálu frekvenčních pásem. My pracujeme na pásmu B20. V Evropské unii se využívají jen tři frekvenční pásma pro provoz sítě NB-IoT a to B3, B8, B20. Nicméně v České republice existují tři operátoři poskytující připojení k této síti, O2, Vodafone a T-Mobile. My jsme si zvolili za operátora Vodafone, který využívá frekvenční pásmo B20. Důležitá přednost modulu je, že má ultra nízkou spotřebu energie. Stačí, aby byl napájen poměrně nízkým napětím 2,2 – 4,3 V a v power saving modu je jeho typická spotřeba pouze 800 nA.

3.3.2.2 Použití v projektu

V našem projektu tento modul využíváme k tomu, aby box mohl komunikovat s webovou aplikací. Modul je připojen přes filtrační kondenzátory k napájecímu napětí 3,3 V. Ke kondenzátorům je také přidaná TVS dioda, aby se zlepšila odolnost proti přepětí. Aby se mohlo s modulem lépe pracovat jsou k modulu připojená dvě tlačítka, jedno je pro funkci „RESET“ a druhý pro funkci „BOOT“. „RESET“ nebo „BOOT“ se aktivuje, pokud piny přivedeme do logické nuly. To můžeme provést buď manuálně, a to zmáčknutím jednoho z tlačítek, anebo softwarově pomocí mikrokontrolerů, a to přes mosfety. Modul komunikuje s mikrokontrolerem přes sběrnici UART. Jelikož pracujeme na úrovni napětí 3,3 V, bylo ještě potřeba uzemnit VIO_SEL pin. Protože modulu umí pracovat buď na 1,8 nebo 3,3 V. Dále je k modulu připojena přes mosfet LED dioda pro signalizaci. Aby se modul mohl identifikovat v síti NB - IOT, je k modulu ještě připojený konektor na SIM kartu. Mezi modulem a konektorem na SIM kartu jsou ještě připojeny 22 Ω rezistory, aby potlačili nepotřebné rušení a zvýšili ochranu proti elektrostatickým výbojům.



Obrázek 11: NB - IOT modul, dostupné z <https://www.soselectronic.com/cz/articles/quectel/vicepasmovy-lte-cat-nb2-modul-bc660k-gl-2593>



Obrázek 12: Schéma NB - IOT modulu

3.3.3 CMI-9653S-SMT-TR | CUI Devices

3.3.3.1 Základní informace

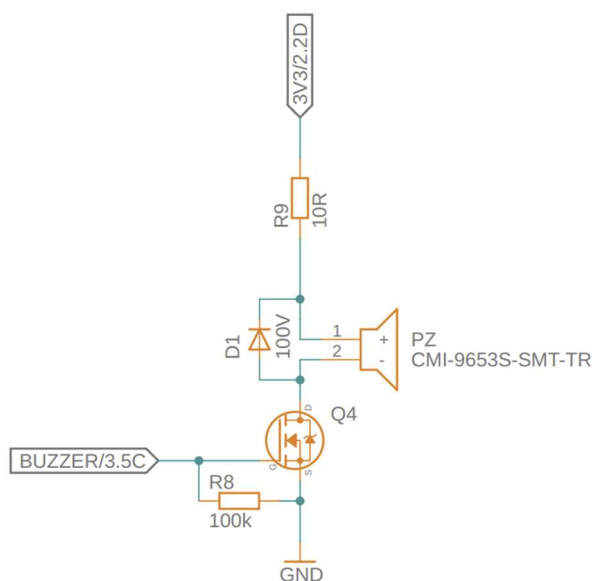
Je to vnitřně řízený elektromagnetický bzučák s hladinou zvuku 80 dB od firmy CUI Devices. Pracuje při napětí od 2 V do 5 V. Při napětí 3 V má spotřebu 30 mA. Typická frekvence zvuku je 2700 Hz. Má dedikovaný oscilátor, takže stačí přivést konstantní ss napětí.⁶

3.3.3.2 Použití v projektu

Elektromagnetický bzučák je MOSFET-N připojen k zemi. Antiparalelně k bzučáku je zapojena flyback dioda se závěrným napětím 75 V a propustným proudem 250 mA, která pomáhá zabránit rychlému vybíjení cívky v bzučáku, a tím zabránit vzniku vysokého napětí a zničení MOSFETu.

3.3.3.3 Princip

Elektromagnetický bzučák převádí střídavé proudy na akustické vlny. Elektrický proud vyvolá mechanické kmity, které se přemění na akustický signál. Využívá se v rozmezí 300 Hz až 3,4 kHz. Pracuje na principu elektromagnetu. Využívá síly, která vznikne v kotvě při průchodu proudu závitů cívky. Jedná se o přímo vyzařující bzučák, což znamená, že akustická energie je vyzařována kmitající membránou navazující bezprostředně na prostředí, do něhož je energie přenášena. Základními parametry reproduktoru jsou účinnost a citlivost. Účinnost je poměr vyzářeného akustického výkonu k elektrickému příkonu, citlivost je dána průměrným akustickým tlakem v ose reproduktoru ve vzdálenosti 1 m a příkonu 1 VA. Důležitou vlastností reproduktoru je směrová charakteristika. Ta udává závislost akustického tlaku před reproduktorem na úhlu, který svírá osa reproduktoru a spojnice reproduktoru a měřicího mikrofону.



Obrázek 13: Schéma zapojení magnetického bzučáku



Obrázek 14: Magnetický bzučák, dostupné z <https://cz.mouser.com/ProductDetail/CUI-Devices/CMI-9653S-SMT-TR>

⁶ (CUI Devices, 2019)

3.3.4 Kapacitní senzory

Kapacitní senzory, nejsou jako ostatní SMD součástky, která by se pájeli přímo na desku plošných spojů. Jsou totiž implementováni přímo v čtyřvrstvé desce plošných spojů. Jak už bylo zmíněno, nakonec jsme se rozhodli využít periférie PTC od firmy Microchip, kterou podporuje i náš mikrokontrolér. Takže byly navrženy do desky plošných spojů kapacitní senzory, které jsou inspirované technologií Qtouch, která je také od Microchipu.

Kapacitní senzory jsou bezkontaktní. Tyto snímače detekují změnu kapacity způsobenou předmětem, který vstoupí do elektrického pole kondenzátoru. To znamená, že jsou schopny rozpoznat nejen vodivé, ale i nevodivé materiály s dostatečně vysokou permitivitou. Díky této vlastnosti dokáží detekovat kovy, plasty i kapaliny a další jiné materiály. My se těmito senzory snažíme detekovat lékařské prášky.⁷

Microchip QTouch je technologie pro dotykové ovládání, která umožňuje implementaci dotykových senzorů v elektronických zařízeních.

Peripheral Touch Controller (PTC) je speciální periférie určená k detekci dotyku na kapacitních senzorech. Externí kapacitní dotykový senzor je obvykle umístěn na desce plošného spoje. Elektrody senzoru jsou přímo připojeny k analogovému vstupu PTC prostřednictvím vstupních/výstupních pinů na zařízení. Umožňuje primárně dva způsoby detekce a to detekování vzájemnou kapacitou a detekcí vlastní kapacitou rozdíl spočívá v tom, jak senzory zachycují dotyk a jak interpretují vstupy.⁸

Detekce vzájemnou kapacitou, anglicky mutual capacitance: Tato metoda využívá dvou vrstev kapacitních senzorů umístěných proti sobě. Jedna vrstva generuje proměnné elektrické pole, zatímco druhá vrstva měří změny tohoto pole způsobené přiblížením vodiče (například prstu). Když se prst přiblíží k povrchu, dojde ke změně kapacity mezi těmito vrstvami, což indikuje dotyk.

Detekce vlastní kapacitou (self capacitance): Tato metoda využívá jedné vrstvy kapacitních senzorů. Tato vrstva je citlivá na změny kapacity v důsledku přiblížení vodiče. Když se prst přiblíží k povrchu, dojde ke změně kapacity této vrstvy, což indikuje dotyk.⁹

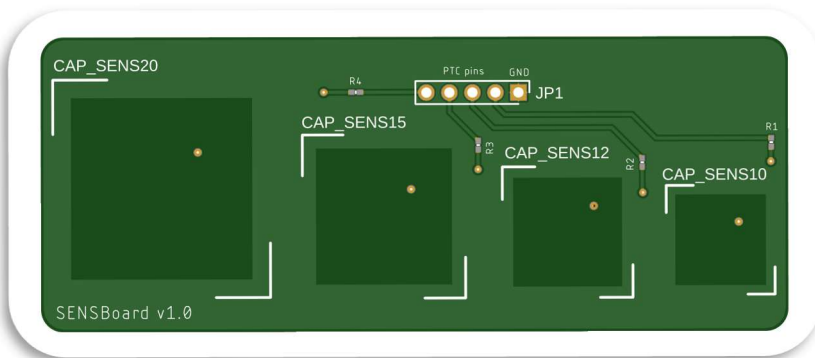
Nám se po delším testování jednotlivých metod detekování prášků nejvíce osvědčilo metoda s vlastní kapacitou.

⁷ (Výukový portál COPtel, 2019)

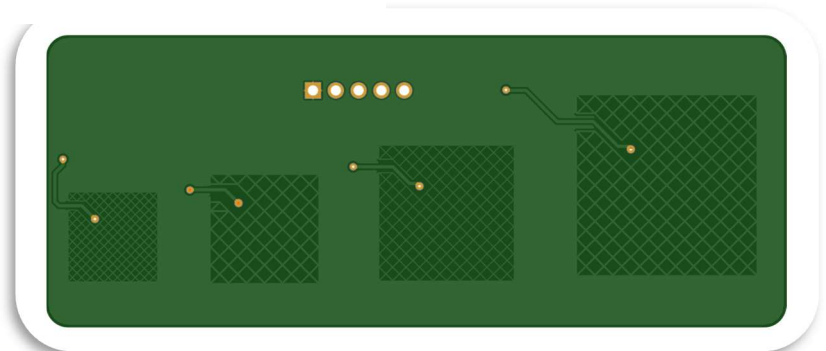
⁸ (Microchip Technology Inc., 2020)

⁹ (Microchip Technology Inc., 2023)

Protože jsme si nebyli jisti zda lékařské prášky budou mít dostatečně vysokou permitivitu a jestli vůbec se nám je podaří detekovat, byla navržena naše vlastní prototypová deska plošných spojů, na které bylo možné jednotlivé senzory testovat.

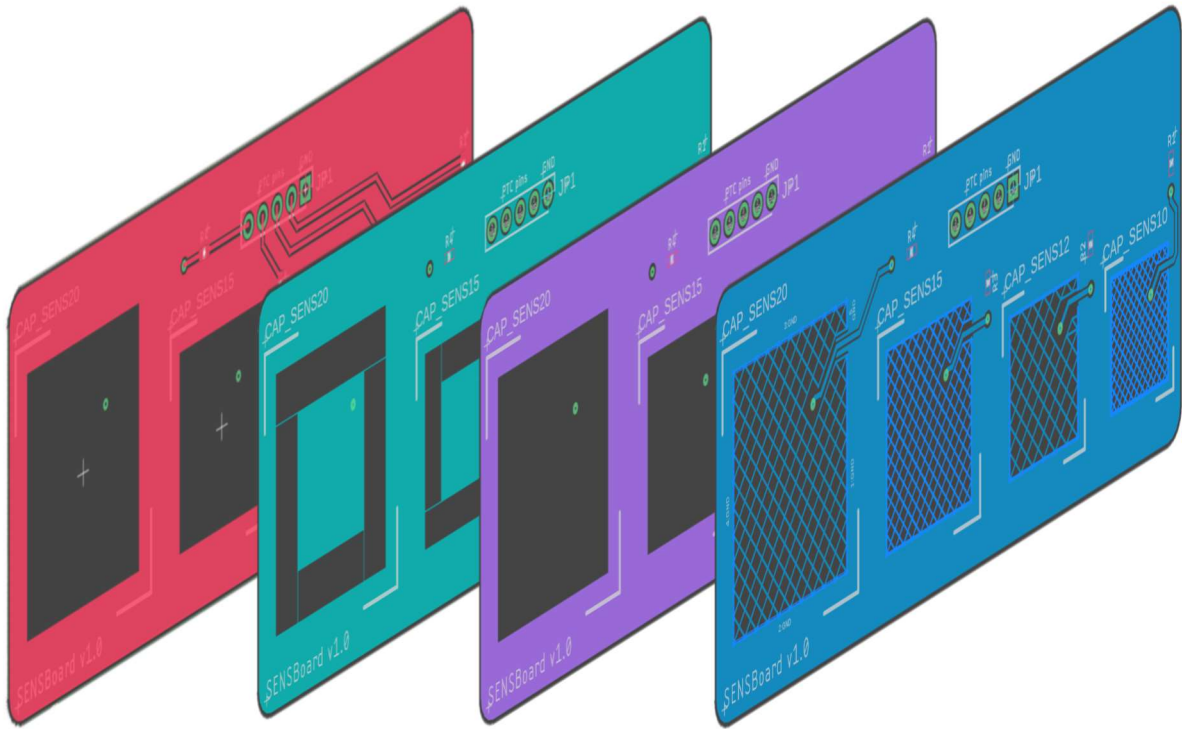


Obrázek 15: Prototypovací deska pro kapacitní senzory



Obrázek 16: Druhá strana prototypovací desky pro kapacitní senzory

Microchip veřejně poskytuje právě pod názvem Qtouch, různé knihovny a příručky jak senzory správně designovat. Tím byla právě deska inspirována. Na obrázku jsou vidět čtyři čtvercové senzory vedle sebe. Každý má jinou velikost, z toho důvodu, abychom mohli testovat jednotlivou citlivost každé velikosti senzoru zvlášť.



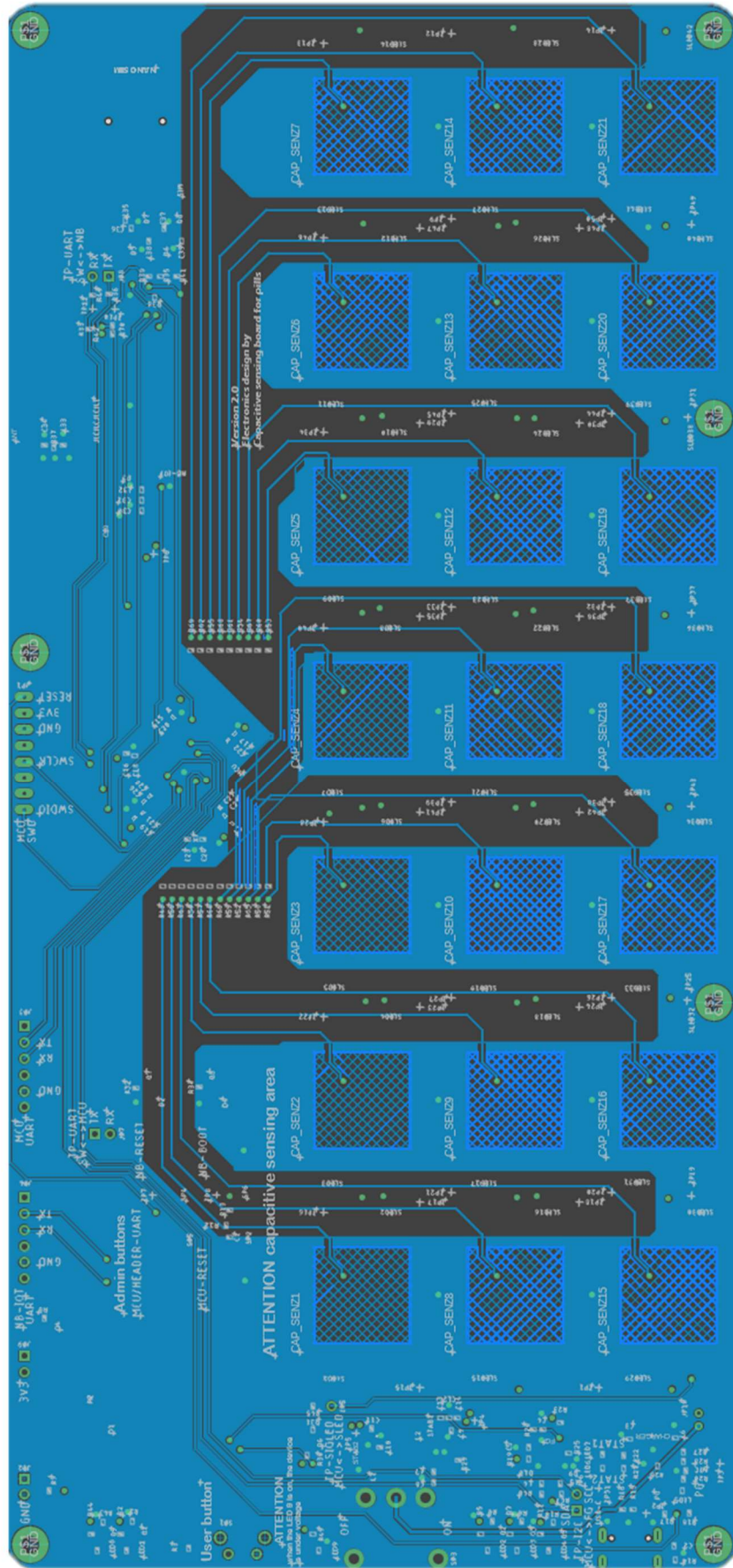
Obrázek 17: Jednotlivé vrstvy prototypovací desky pro kapacitní senzory

Když uvažujeme samotný senzor, tak ten se rozkládá ve druhé a čtvrté vrstvě desky plošných spojů. V první vrstvě, jak je vidět na obrázku je měď nad senzorem vyříznutá, protože jinak je celá deska zalitá v mědi a jsou v ní jen odizolované určité cesty. Ve druhé vrstvě je samotný senzor, ten představuje ta měděná plocha uprostřed, která je odizolovaná od desky větší mezerou. Ve třetí vrstvě je zase měď vyříznutá ve tvaru senzoru stejně jako je tomu ve vrstvě první. A ve čtvrté vrstvě desky plošných spojů je udělané stínění, které připomíná tvar mřížky. Ta je tam aby se zamezilo případným rušením a nežádoucím detekováním senzoru zespoda desky plošných spojů. Samotná signálová cesta senzoru vede prokovem do čtvrté vrstvy a dále je vedena až ke kolíkové liště, ve finální verzi k mikrokontroleru kde je vyvedena do první vrstvy. Je to z toho důvodu, protože kdyby rovnou vedla v první vrstvě mohlo by docházet k náhodným dotykům. Dále je vedena na rezistor, který je doporučený do cesty dát pro útlum zarušení signálu a zlepšení následného detekování.

Hodnota rezistoru není pevně daná a doporučuje se senzor otestovat s různými hodnotami odporu zhruba od 0Ω - $10k\Omega$, v extrémních případech je možnost osadit až $100k\Omega$ nebo dokonce $1M\Omega$. Pak je ale následně třeba provést určité změny v softwarové části. Nám se nevíce osvědčila hodnota odporu $1k\Omega$.¹⁰

Jelikož se při testování tento typ senzoru poměrně osvědčil, je také použit s mírnými úpravami velikosti (kvůli citlivosti) ve výsledné verzi desky plošných spojů.

¹⁰ (Microchip Technology Inc., 2014)



Obrázek 18: Finální deska plošných spojů, konkrétně 4 vrstva, stínění kapacitních senzorů

3.3.5 Akumulátor L584070 | CELLEVIA BATTERIES

3.3.5.1 Základní informace

Jedná se o Polymer Lithium-Ion baterii s kapacitou 2400 mAh a jmenovitým napětím 3,7 V. Má výdrž 300 nabíjecích cyklů. Je vyrobena firmou CELLEVIA BATTERIES.

3.3.5.2 Princip

Elektrochemické zdroje jsou zdrojem elektrické energie, která vzniká v průběhu chemické oxidačně redukční reakce. Při ní oxidující se látka ztrácí elektrony, redukující se látka elektrony získává. Skládají se z jednoho nebo několika galvanických článků. Každý článek poskytuje napětí obvykle od 0,5 V do 4 V. To je určeno rozdílem stykových potenciálů materiálů elektrod. Galvanický článek se skládá ze dvou elektrod (kladné a záporné) ponořených do elektrolytu. Elektrody jsou obvykle kovové desky nebo mřížky z různých materiálů pokryté reaktanty. Na jedné straně je oxidační činidlo, na druhé redukční činidlo. Elektrolyt je kapalina nebo tuhá fáze s iontovou vodivostí. Ponoří-li se elektrody do elektrolytu, ustaví se mezi nimi rozdíl potenciálů nazývaný bezproudové napětí. Po připojení vnějšího vodivého obvodu přecházejí elektrony ze záporné elektrody ke kladné. Současně začnou na povrchu ponořených elektrod probíhat elektrochemické reakce, které produkují nebo spotřebovávají elektrony. Elektrony jsou produkovány při oxidaci a spotřebovávány při redukci. Při redukci v blízkosti kladné elektrody jsou produkovány ionty, které jsou transportovány elektrolytem k záporné elektrodě a vstupují do oxidace. Tím vzniká uzavřený elektrický obvod, jímž prochází stejný elektrický proud. Při rozpojení vnějšího obvodu nebo spotřebování některého z reaktantů se obě elektrochemické reakce přeruší a tím zanikne i proud. Dělí se na primární a sekundární články. U primárních článků jsou proudotvorné chemické reakce nevratné. Napětí vybitého článku se nedá obnovit. Sekundárním článkem může vlivem vnějšího napětí procházet proud i opačným směrem. Elektrodové reakce jsou v tomto případě vratné. Elektrická energie se přeměňuje na chemickou, dochází k elektrolýze a článek se nabíjí.¹¹

Sekundární článek je po vybití možné znovu nabít. Při nabíjení hromadí elektrickou energii a poté ji zpětným elektrochemickým pochodem vydává. Zatímco při vybíjení článku je anodou záporná elektroda a katodou kladná elektroda, při nabíjení je tomu naopak. Pro nabíjení se používá stejnosměrný nepřetržitý proud, tepavý proud nebo impulzní proud s proměnnou střídou. Tepavý nebo impulzní proud zlepšuje uložení dodávané energie do aktivního materiálu elektrod.



Obrázek 19: Baterie Li - Pol, dostupné z <https://www.ampul.eu/cs/baterie/4694-li-pol-baterie-4000mah>

¹¹ (Wikipedie, otevřená encyklopedie, 2024)

3.3.6 IN-PI42TASPRGPB | Inolux

3.3.6.1 Výběr RGB diody

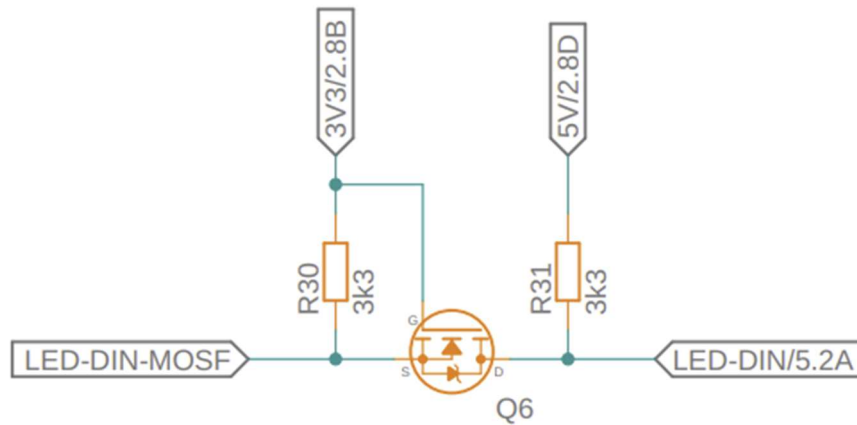
Stav každého políčka na léky indikuje dvojice barevných LED diod. Celkový počet políček na léky je dvacet jedna. U každého políčka jsou dvě indikační diody aby políčko bylo nasvíceno souměrně a nedocházelo tak tomu, že by uživatel nevěděl k jakému políčku daná LED dioda svítí. Indikačních LED diod je tedy čtyřicet dva. Počet LED, které potřebujeme nezávisle ovládat, je relativně dost veliký a i při maticovém uspořádání 3 x 7 bude třeba využít alespoň deset logických výstupů MCU nebo přidat dekodér 1 z 8mi.

Elegantní řešení nabízí americká firma Inolux. Jedná se o čtyř vývodový integrovaný obvod obsahující na čipu RGB LED a tři osmibitové registry pro uložení odstínů základních barev. Kromě napájecích přívodů obsahuje Din a Dout vývody a obvody je možné řadit do kaskády a vytvářet relativně dlouhé řetězce diod. Rychlost sériového přenosu dat je relativně vysoká 800 kbit za sekundu. Logický vstup a výstup je kompatibilní s řadou logických integrovaných obvodů 74 HC, která má poměrně vysokou šumovou imunitu, a tedy vzdálenost mezi jednotlivými LED čipy může být relativně velká.¹²

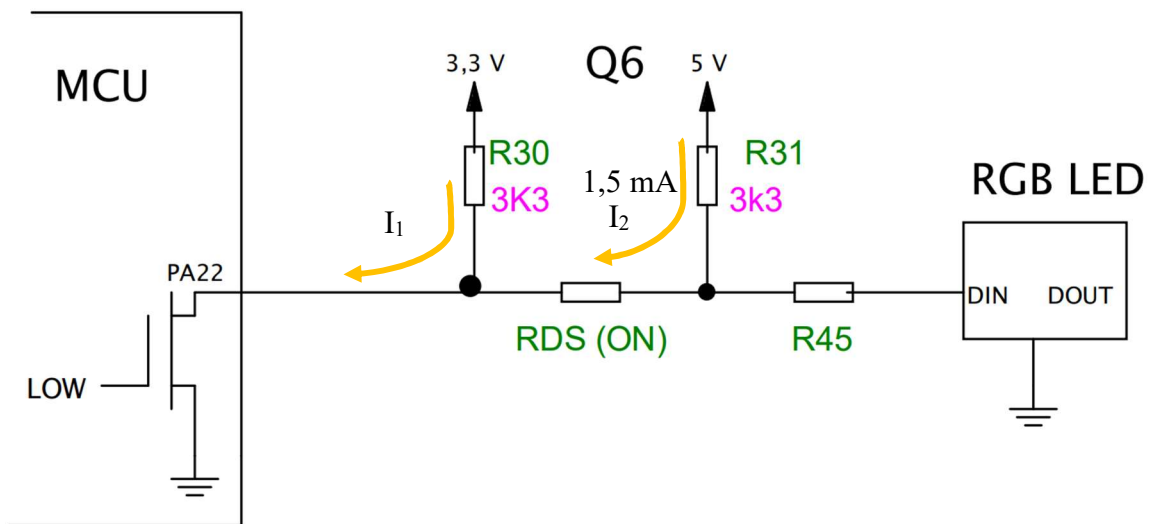
3.3.6.2 Napojení LED řetězce na MCU kontrolér

Náš MCU kontrolér je kvůli minimalizaci spotřeby energie napájen z 3,3 V zdroje. Přímé napojení RGB LED řetězce na MCU tedy kvůli nízké úrovni log 1 na výstupu MCU není možné. Problém je možné řešit pomocí speciálního integrovaného obvodu (např. SN74LVC1T45). Vzhledem k tomu, že se jedná o jeden logický výstup z MCU bude ekonomicky výhodnější konverzi úrovní řešit tranzistorem Q6 – MOSFET typu N s indukovaným kanálem. Je však třeba správně nastavit pracovní bod tranzistoru výpočtem vhodného rezistoru. Bude-li mít rezistor R30 příliš malý odpor, může dojít v log 0 k přetížení výstupního tranzistoru MCU. Pull up rezistor R31 zajistí dostatečné napětí v log 1 na vstupu Din RGB čipu.

¹² (Inolux Corporation, 2018)



Obrázek 20: Schéma převodníku napětí k programovatelným LED diodám



Obrázek 21: Podrobně rozkreslené schéma funkce převodníku napětí

Rezistory R30, R31 zvolíme $R30 = R31 = 3k3\Omega$ a provedeme kontrolní výpočet proudu tekoucího při log 0 tranzistorem výstupu PA22 kontroléru MCU.

Výpočet proudu:

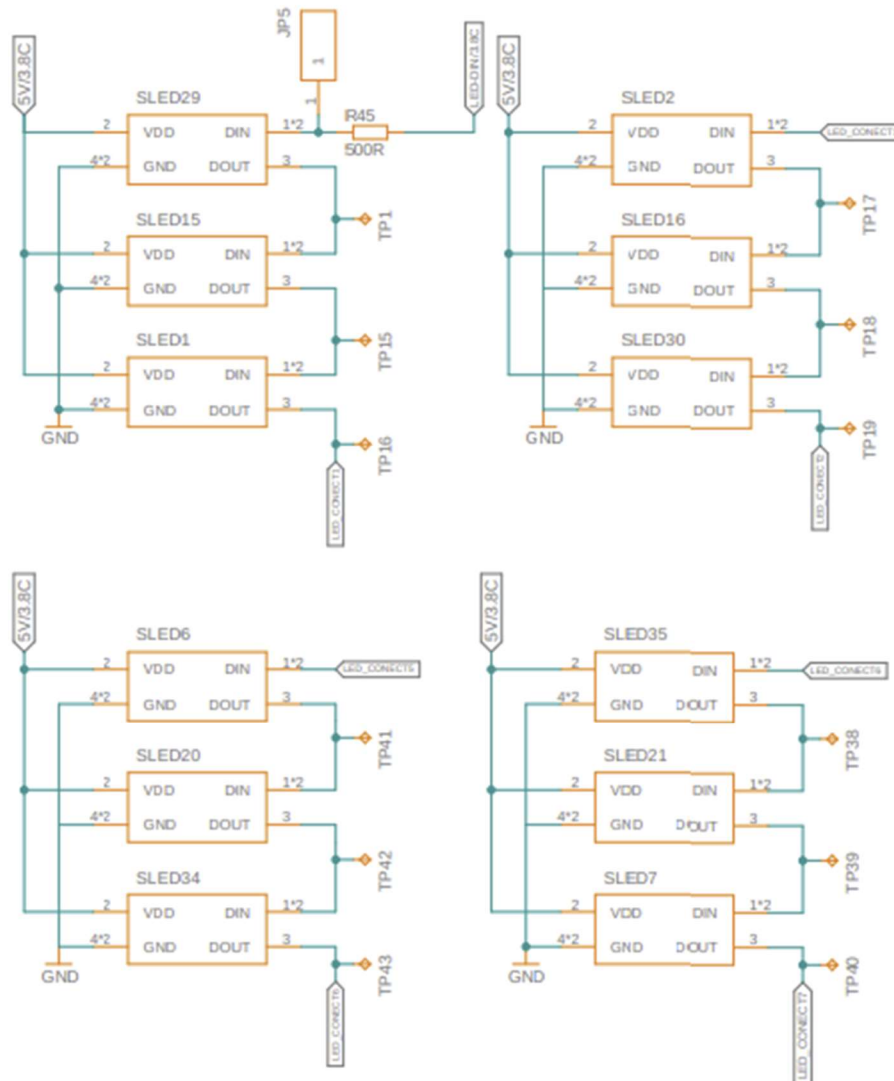
$$I_1 = \frac{3,3}{3,3 * 10^3} = 1 \text{ mA}$$

$$I_2 = \frac{5}{3,3 * 10^3} = 1,5 \text{ mA}$$

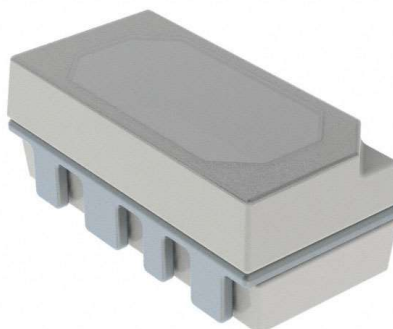
$$I = I_1 + I_2 = 1 + 1,5 = 2,5 \text{ mA}$$

Rovnice 4: Výpočet proudu tekoucího při log 0 tranzistorem výstupu PA22

Bude-li logický výstup PA22 kontroléru v log 0, tranzistor Q9 bude v saturaci a odpor vodi-vého kanálu $R_{DS(on)}$ bude jednotky ohmů. Při nastavení pull up rezistorů $R30 = R31 = 3k3\Omega$ bude výstupním tranzistorem protékat proud cca 2,5 mA, což je přijatelné.



Obrázek 22: Ukázka zapojení 12 z 42 programovatelných RGB LEDek



Obrázek 23: RGB programovatelné LED, dostupné z <https://www.digikey.com/en/products/detail/inolux/IN-PI42TASPRGPB/9681236>

3.3.7 MCP73833T | Microchip

3.3.7.1 Základní informace

Je řadič řízení lineárního nabíjení od firmy Microchip, který dokáže nabíjet baterie až 1200 mA a je určen pro nabíjení jedno článkové Lithium-Ion a Lithium-Polymer baterie, a to při různých úrovních napětí 4.2V, 4.35V, 4.4V nebo 4.5V. Charger dokáže regulovat nabíjecí proud na základě teploty, protože je k němu připojen termistor NTC. Tepelná regulace optimalizuje dobu nabíjecího cyklu.

3.3.7.2 Použití v projektu

Na vstup a výstup jsou připojeny bypassing kondenzátory 4,7 μ F. Pomocí rezistoru R₂₆ je nastaven nabíjecí proud. Nabíjecí napětí jsme nastavili na 4,2 V a proud rychlého nabíjení na 500 mA, který jsme určili podle datasheetu baterie. Rezistor nastavující nabíjecí proud jsme vypočítali na hodnotu 2 k Ω . Rezistor na nastavení proudu jsme vypočítali podle vzorce:¹³

$$R_{PROG} [k\Omega] = \frac{1000 V}{I_{REG} [mA]}$$

$$R_{PROG} = \frac{1000}{500} = 2 k\Omega$$

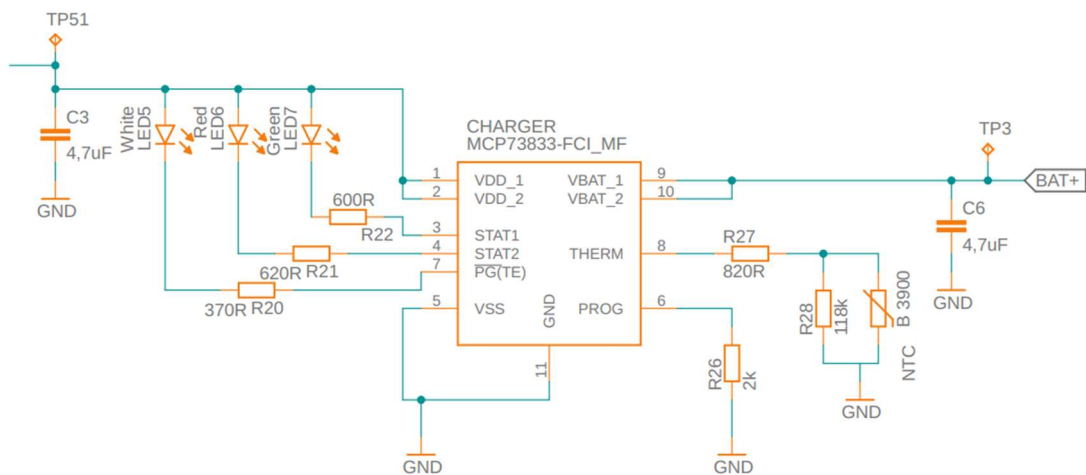
Rovnice 5: Výpočet rezistoru pro nastavení proudu

NTC termistor hlídá teplotu a při příliš vysoké nebo nízké teplotě charger přestane nabíjet. Hodnota termistoru je výrobcem určena na 10 k Ω . Na signalizaci stavu nabíjení jsou k chargeru připojeny tři LED diody, každá z jednotlivých tří LED diod je připojena na jeden signální pin chargeru, zelená je připojena na pin STAT1, červená na pin STAT2 a bílá na pin PG (POWER GOOD), tyto piny podle stavů ve kterých jsou signalizují podle tabulky umístěné níže určitý stav ve kterém se charger právě nachází. K LED diodám jsou také sériově připojeny odpory, které jsou vypočítány pomocí Ohmova zákona:

$$R = \frac{U}{I}$$

Rovnice 6: Ohmův zákon

¹³ (Microchip Technology Inc., 2009)



Obrázek 24: Schéma zapojení CHARGERU

Charge Cycle State	STAT1	STAT2	PG
Shutdown	Hi-Z	Hi-Z	Hi-Z
Standby	Hi-Z	Hi-Z	L
Charge in Progress	L	Hi-Z	L
Charge Complete (EOC)	Hi-Z	L	L
Temperature Fault	Hi-Z	Hi-Z	L
Timer Fault	Hi-Z	Hi-Z	L
System Test Mode	L	L	L

Obrázek 25: Tabulka možných stavů chargeru, dostupné z https://cz.mouser.com/datasheet/2/268/MCHPS02791_1-2520625.pdf

3.3.7.3 Princip

Lineární charger využívá regulovatelného lineárního stabilizátoru. Výstupní napětí (nebo jeho část) je porovnáváno s referenčním pomocí zesilovače odchylky. Při zjištění změny výstupního napětí je ovládán regulační prvek. Při poklesu výstupního napětí se regulační prvek více otevře, sníží se na něm úbytek napětí a zvýší se napětí na zátěži. Při zvýšení úbytku napětí na zátěži se regulační prvek více uzavře, zvýší se na něm úbytek napětí a sníží se napětí na zátěži. Výhodou lineárního chargeru jsou malé rozměry a malé potřebné množství součástek, nevýhodou je velká výkonová ztráta v chargeru. Charger dobíjí baterii pouze pokud je teplota v rozmezí 0 °C až 55 °C. Nabíjecí napětí a proud rychlého nabíjení jsou nastaveny pomocí externích rezistorů.

3.4 STC3115 | STMicroelectronics

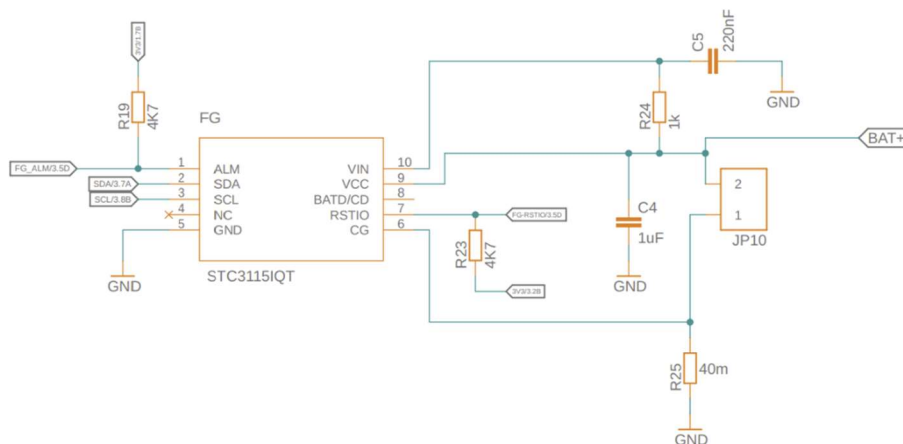
3.4.1 Základní informace

STC3115 je obvod typu Fuel Gauge pro monitorování stavu baterie od firmy STMicroelectronics. Poskytuje kontrolu stavu jednočlánkových baterií. Je napájen přímo z baterie. S MCU komunikuje přes I2C sběrnici a přerušení (interrupt) - Alarm a Reset.

3.4.1.1 Popis funkce a zapojení

STC3115 je vlastně specializovaný kontrolér, který periodicky vzorkuje tři analogové vstupy – napětí, proud a teplotu baterie, porovnává je s přednastavenými hodnotami charakterizujícími konkrétní typ monitorované baterie a počítá state-of-charge (SOC) = aktuální stav nabití baterie. Napětí baterie je přivedeno na pin VIN. Výrobce obvodu zde doporučuje osadit filtr tvořený rezistorem R24 a kondenzátorem C5. Součin jejich hodnot určuje časovou konstantu obvodu. Proud baterie je měřen nepřímo jako úbytek na bočníku R25. Jako bočník je použit přesný rezistor (tolerance 1%) $R25 = 40\text{m}\Omega$. Pro teplotní kompenzaci hodnot je na čipu osazen teplotní senzor. Na základě naměřených hodnot proudu a napětí v reálném čase jsou vyhodnocovány přírůstky nebo úbytky elektrického náboje a aktuální stav baterie. Přes I2C sběrnici jsou aktuální hodnoty vyčítány MCU. Výstupní alarm signalizuje stav nízkého SOC a může také indikovat nízké napětí baterie. Prahové úrovně alarmu jsou programovatelné. Piny Alarm a Reset jsou typu „open drain“ a je třeba osadit rezistory R19 a R23. STC3115 nabízí pokročilé funkce pro zajištění vysokého výkonu měření stavu nabití za všech podmínek aplikace.

nízké napětí baterie. Prahové úrovně alarmu jsou programovatelné. Piny Alarm a Reset jsou typu „open drain“ a je třeba osadit rezistory R19 a R23. STC3115 nabízí pokročilé funkce pro zajištění vysokého výkonu měření stavu nabití za všech podmínek aplikace.¹⁴



Obrázek 26: Schéma zapojení Fuel Gauge

¹⁴ (STMicroelectronics, 2018)

3.4.2 TPS63021DSJT | Texas Instruments

3.4.2.1 Základní informace

Je to Buck-boost spínaný stabilizátor s jednou cívkou od firmy Texas Instruments. Pracuje se vstupním napětím od 1,8 V do 5,5 V. Má pevné výstupní napětí 3,3 V a maximální výstupní proud 4 A. Napětí na výstupu monitoruje přes vstup FB. K dispozici je také power good výstup. Pracuje s přepínací frekvencí 2,4 MHz. Pro správnou funkci může být baterie vybita pod 2 V.

3.4.2.2 Použití v projektu

Na vstup jsme připojili skupinovou kapacitu dvou kondenzátorů o kapacitách 10 μF a na výstup skupinovou kapacitu tří kondenzátorů o kapacitách 22 μF pro vyhlazení napájení. Enable vstup je připojen k napájení, aby byl stabilizátor aktivní. Přes 1M Ω pull-down rezistor je vstup PS/SYNC připojen ke GND, což znamená, že je aktivní power save mode. Power save mode se zapne, pokud proudový odběr klesne pod 100 mA. Napájecí vstup pro interní kontrolní obvody VINA jsme připojili přes 100nF kondenzátor do GND, aby byly stabilně napájeny nízko šumovým napětím. Na piny L1 a L2 jsme připojili cívku s hodnotou 1 μH . Cívka by měla mít co nejmenší stejnosměrný odpor, aby se zabránilo ztrátám na cívce. Minimální saturační proud cívky jsme vypočítali pomocí vzorce:

$$I_{PEAK} = \frac{I_{OUT}}{\eta * (1-D)} + \frac{V_{IN} * D}{2 * f * L}$$

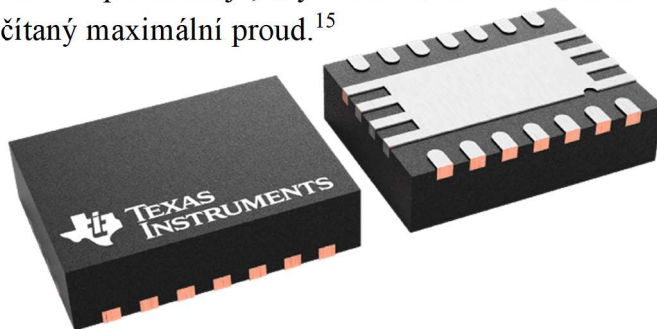
Rovnice 7: Vzorec pro výpočet minimálního saturačního proudu cívky

D je délka trvání v boost modu a vypočítá se:

$$D = \frac{V_{OUT} - V_{IN}}{V_{OUT}}$$

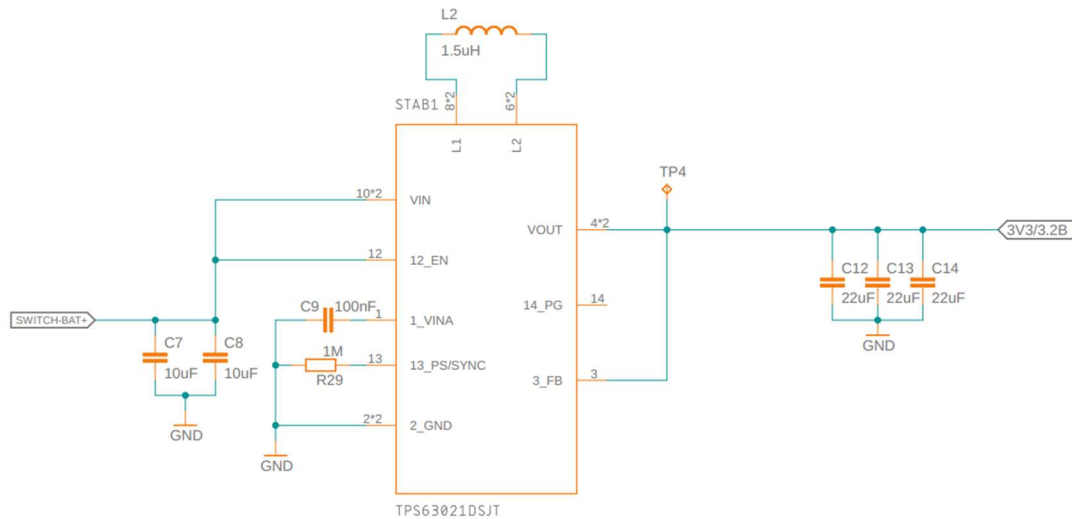
Rovnice 8

a η je odhadovaná konverzní účinnost, kterou v datovém listu uvádějí jako předpoklad 0,9. Očekáváme maximální odebíraný proud 300 mA a vypočítali jsme minimální saturační proud cívky 207,6 mA. Doporučené je, aby měla cívka minimálně o 20 % vyšší saturační proud, než je vypočítaný maximální proud.¹⁵



Obrázek 27: Stabilizátor TPS63021DSJT, dostupné z: <https://www.ti.com/product/TPS63021/part-details/TPS63021DSJT>

¹⁵ (Texas Instruments Incorporated, 2019)

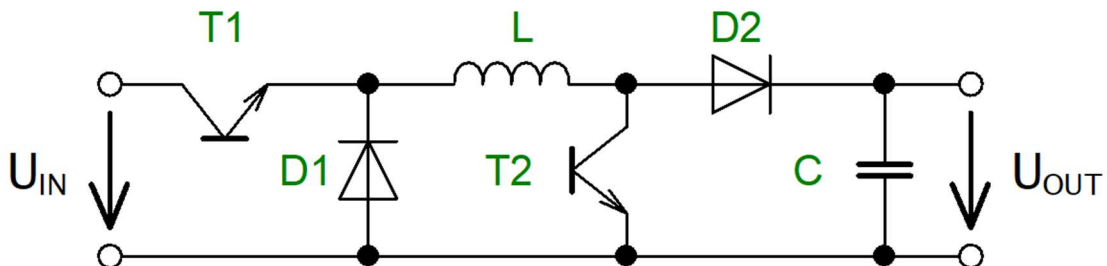


Obrázek 28: Schéma zapojení stabilizátoru na 3,3 V

3.4.2.3 Princip

Regulační tranzistor je ve spínacím režimu (sepnuto/vypnuto). Přes pin FB je komparátorem s hysterezí porovnáváno výstupní napětí U_{OUT} s referenčním napětím U_{REF} . Na stabilizátoru vzniká minimální výkonová ztráta na rozdíl od lineárního stabilizátoru. V sepnutém režimu je úbytek napětí U_{CE} přibližně 0 V ($P = U \times I$) a v rozepnutém stavu je I_C přibližně 0 A. Malá výkonová ztráta vzniká pouze během sepnutí nebo rozepnutí. Jedná se o buck-boost měnič. Nevýhodou je nutnost zapojení akumulacího prvku cívky a kondenzátoru na výstup, aby se minimalizovalo zvlnění napětí při rozepnutém stavu.

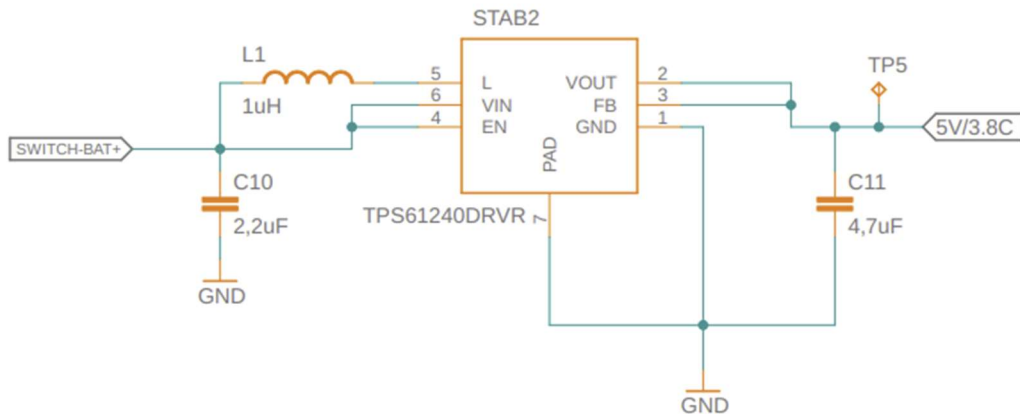
Jak již bylo napsáno výše, jedná se o buck-boost měnič. Buck-boost měnič má dva spínací prvky. Tyto spínací prvky jsou spínané současně. V sepnutém stavu tranzistorů je cívka dobíjena vstupním proudem. V rozepnutém stavu je kondenzátor dobíjen proudem z energie nashromážděné v cívce během sepnutého stavu. Stabilizace U_{OUT} je řízena změnou doby sepnutí a vypnutí. Na výstupu buck-boost měniče může být nižší i vyšší napětí než na vstupu.



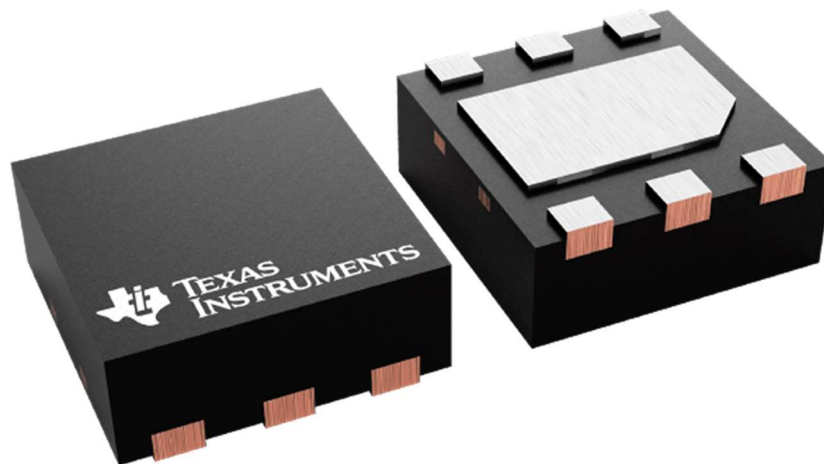
Obrázek 29: Schéma zapojení neinvertujícího buck-boost měniče

3.4.3 Napěťový stabilizátor TPS61240

Napěťový stabilizátor TPS61240 firmy Texas Instruments je vysoce účinný, synchronní, napětí zvyšující DC-DC měnič optimalizovaný pro obvody napájené buď tříčlávkovou alkalickou, NiCd nebo NiMH baterií nebo jednočlávkovou Li-Ion nebo Li-Pol baterií s pevným výstupním napětím $5,0\text{ V} \pm 2\%$ a proudem do 450mA. Rozsah vstupního napětí je 2,3V -5,5V. Účinnost měniče je za standardních podmínek vyšší než 90%. K obvodu stačí připojit doporučenou tlumivku $L1 = 1\mu\text{H}$ a kondenzátory C10 a C11.¹⁶



Obrázek 30: Schéma stabilizátoru na 5V



Obrázek 31: Stabilizátor TPS61240DRVR, dostupné z: <https://www.ti.com/product/TPS61240/part-details/TPS61240DRVR>

¹⁶ (Texas Instruments Incorporated, 2015)

3.4.4 USB C | Global Connector Technology (GCT)

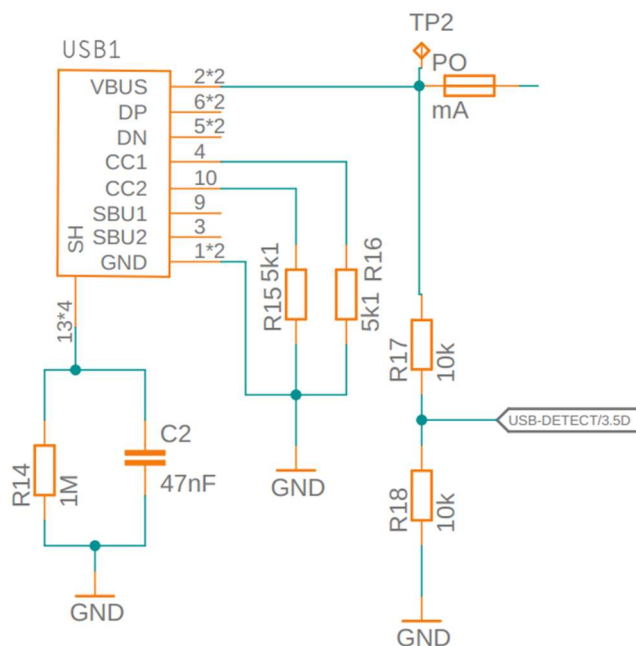
3.4.4.1 Základní informace

USB4105-GF-A-060 je USB C od firmy GCT. Je designované pouze pro nabíjení, a tak přestože má podporu standartu USB 2.0, není pro přenos dat využíváno. Maximální povolené proudy jsou pro power piny až 5 A. Maximální odpor kontaktů je 40 mΩ.

3.4.4.2 Použití v projektu

Jelikož je zařízení vybaveno baterií, aby nemuselo být stále na jednom místě, a dalo se s ním snadno manipulovat. Baterii je samozřejmě potřeba také nějak nabíjet, to je umožněno díky konektoru USB C, kterým box disponuje. Když už uživatel baterii vybijí, může ji jednoduše dobít pomocí klasického USB-C kabelu, se kterým se dnes setkáme nejčastěji. Je to také kvůli tlaku Evropské unie, která stanovila datum zavedení jednotné nabíječky pro všechny mobilní telefony, tablety a další elektroniku. Nové nabíječky budou povinné od prosince roku 2024. Takže se uživateli našeho boxu prakticky nestane, že by se zařízení vybil a uživatel by ho neměl čím dobít. Prostě vezme svůj USB C kabel například od svého telefonu nebo tabletu a jednoduše s ním nabije box.

Aby čip v napájecím zdroji věděl, jaké má mít výstupní napětí, přidali se do schématu na výstupy CC1 a CC2 a rezistory R15 a R16 v hodnotě 5,1kΩ a připojili se k zemi. Takto nastavené piny by měli zajišťovat, že na výstupu bude stabilní napětí 5V a maximální proud bude omezen na 3A. Proud 3A už je poměrně velký proud a proto jsme ještě pro jistotu použili za obvodem pojistku na 700mA, aby nebyl překročen nabíjecí proud baterie.



Obrázek 33: Schéma zapojení USB C konektoru s pojistkou



Obrázek 32: USB C, dostupné z <https://cz.mouser.com/ProductDetail/GCT/USB4105-GF-A-060?qs=QNEbhJQKvY2mowNJzgfA%3D%3D>

3.4.5 Signalizační LED

3.4.5.1 Základní informace

Pro signalizaci jsme použili LED, které svítí do úhlu, konkrétně do úhlu pravého 90°. Používáme převážně diody, od firmy Würth Elektronik a jednu od firmy Inolux. Konkrétně používáme modely Würth Elektronik zelená - 155060GS75300, modrá - 155060BS75300, červená - 155060RS75300, žlutá - 155060YS75300, oranžová -155060AS75300 a Inolux bílá - IN-S63AS5UW.

3.4.5.2 Použití v projektu

V projektu používáme tyto LED diody pro signalizaci různých stavů. Všechny jsou umístěny na levé straně PCB, téměř u hrany, proto byly zvoleny LED diody úhlové. Celkem jich pro signalizaci používáme šest. Všechny jsou připojeny přes ochranný odpor, aby nedošlo k jejich zničení. Ten je vypočítán dole podle rovnice (9). Čtyři jsou ještě zapojeny přes mosfet a jsou tak ovládány z mikrokontroleru, kam jsou přivedeny jako GPIO piny. Tři z nich, konkrétně zelenou oranžovou a červenou LED diodu používáme pro signalizaci stavu baterie nabití. Čtvrtá z nich, žlutá LED dioda je jako záložní pokud by se našel nějaký stav, který by byl potřeba také signalizovat. Pátá modrá LED dioda je také připojena přes mosfet, ale není ovládána pomocí mikrokontroleru, ale je připojena na NB – IOT modulu, aby zde mohla signalizovat různé stavy. Ta poslední, která má bílou barvu je připojena jen přes ochranný odpor rovnou na stabilizátor 3,3 V a signalizuje tak, jestli je zařízení v provozu a pod napětím.

Pomocí Ohmova zákona jsme vypočítali předřadné rezistory.

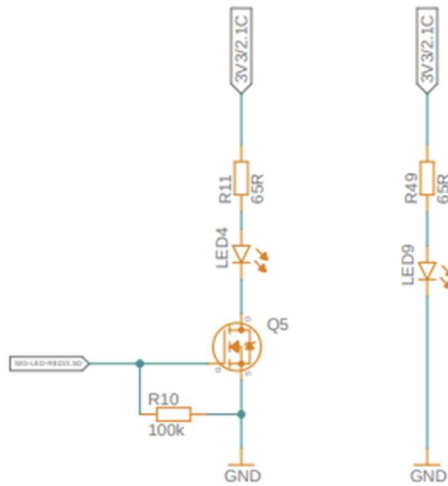
$$R = \frac{U_{3V3} - U_D}{I}$$

$$R = \frac{3,3 - 3,2}{0,02}$$

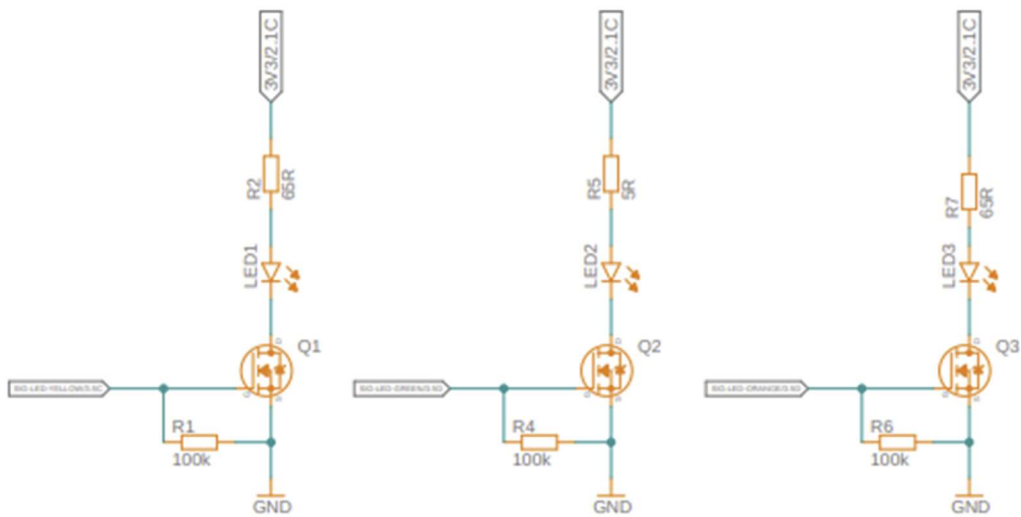
$$R = 5 \Omega$$

Rovnice 9: Výpočet ochranných odporů k LED diodám

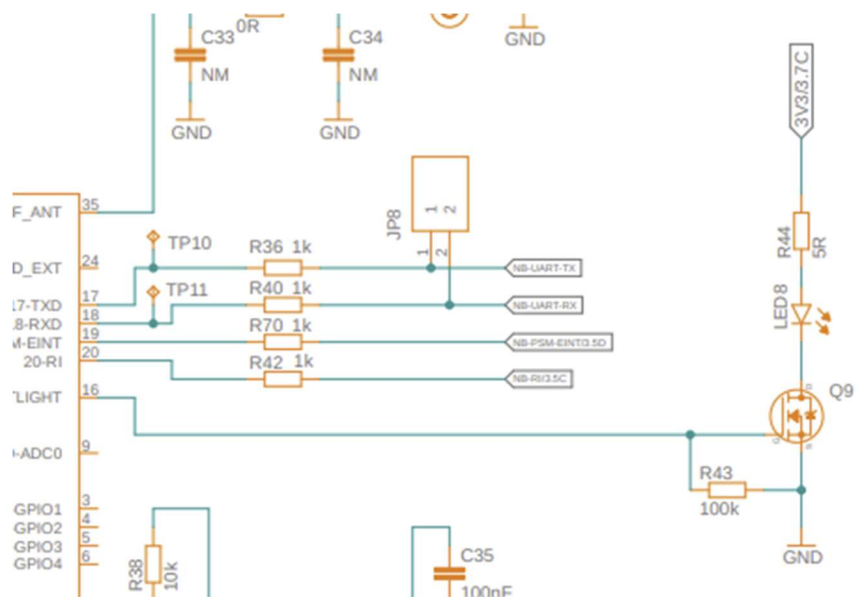
LED jsou přes MOSFET-N připojeny k zemi. Gate je připojen k padu MCU. Mezi gate a source MOSFETu je zapojen rezistor, který slouží pro vybití parazitní kapacity hradla, a tak bezpečnému rozpojení tranzistoru.



Obrázek 35: Signalizační LED diody

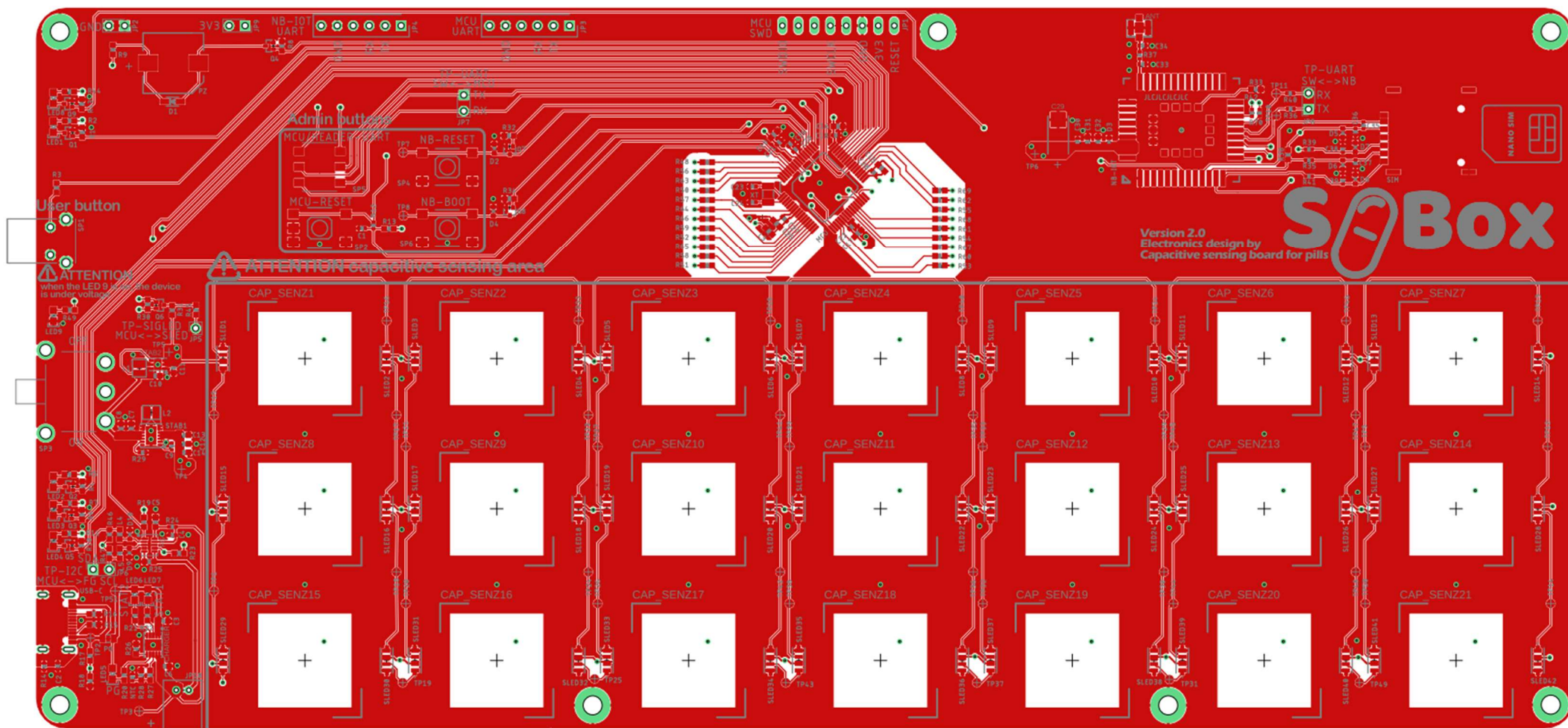


Obrázek 36: Signalizační LED diody ovládané z mikrokontroleru



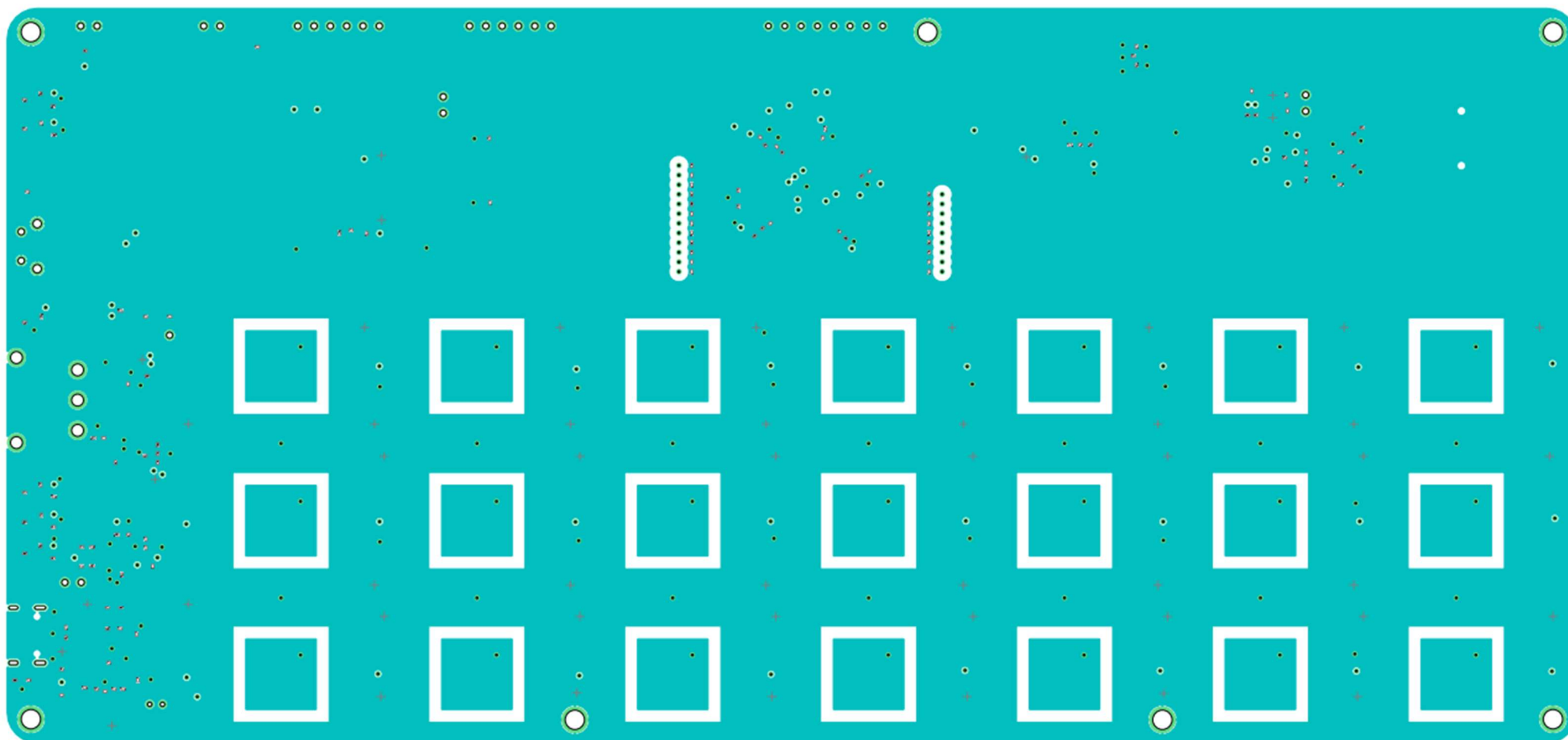
Obrázek 37: Signalizační LED dioda ovládaná z NB - IOT modulu

4 NÁVRH PCB – VRSTVA 1



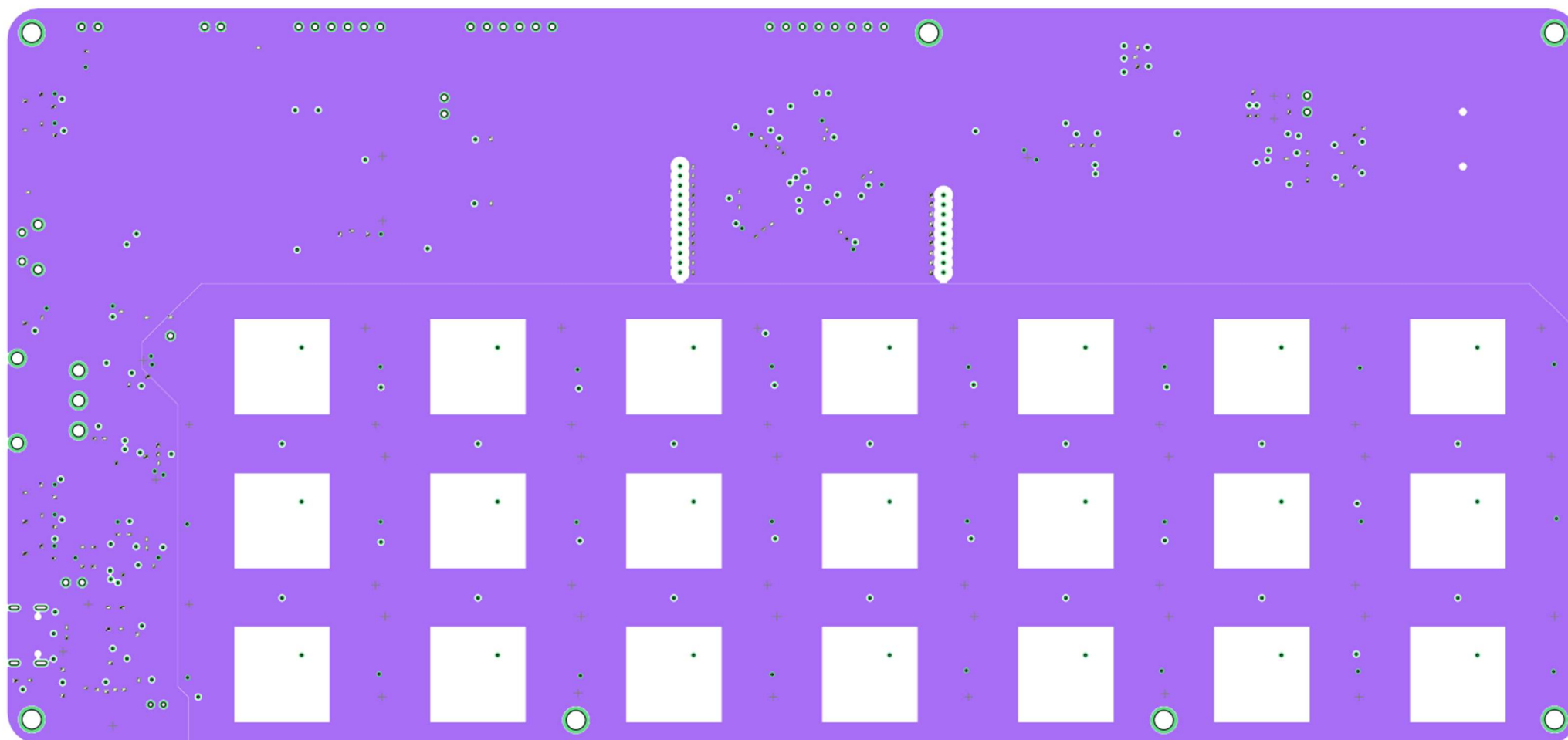
Obrázek 38: 1. vrstvá finální desky

5 NÁVRH PCB – VRSTVA 2



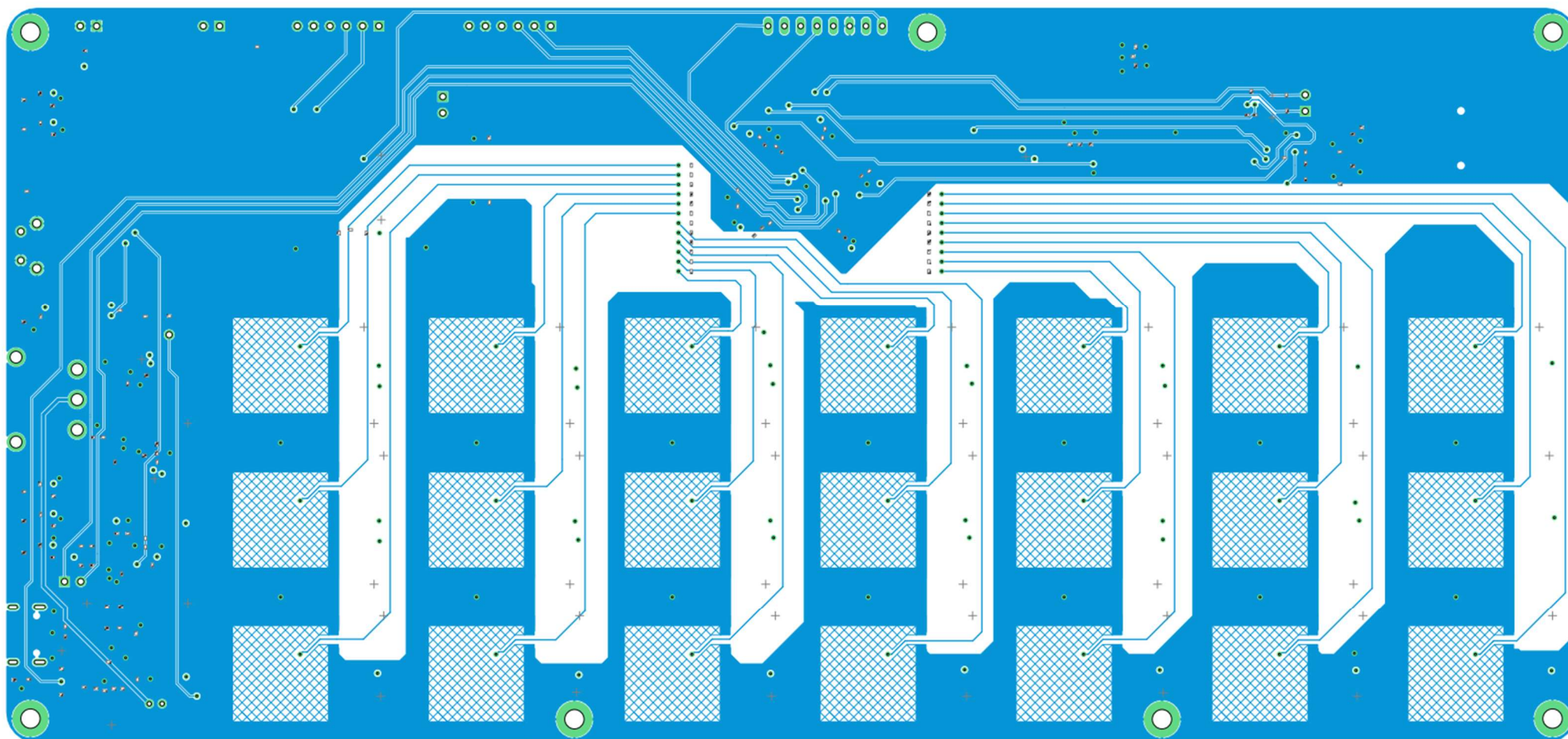
Obrázek 39: 2. vrstva finální desky

6 NÁVRH PCB – VRSTVA 3



Obrázek 40: 3. vrstva finální desky

7 NÁVRH PCB – VRSTVA 4



Obrázek 41: 4. vrstva finální desky

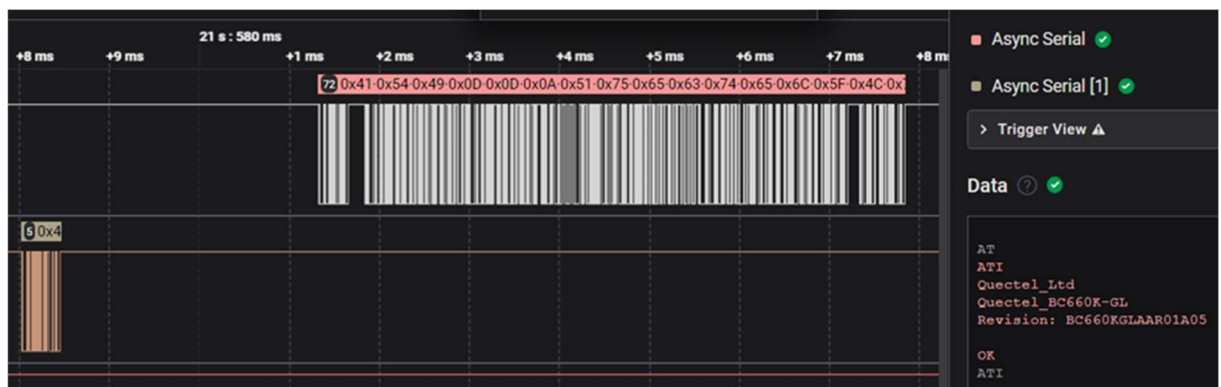
8 VÝROBA, PRŮBĚH OSAZOVÁNÍ A NÁSLEDNĚ VZNIKLÉ CHYBY

Ještě před navržením finální desky plošných spojů, jsme jak už bylo zmíněno objednávali malé prototypovací desky pro kapacitní senzory, protože bylo potřeba nejdříve vyzkoušet jestli budou léky vůbec umět detekovat. Další důvod byla cena, jelikož deska je více jak třikrát větší jak ložský model, výroba finální desky je opravdu drahá. Po návrhu jsme tedy vygenerovali GERBER data. Tyto data jsou důležité pro výrobce desky a je potřeba je mít opravdu správně vygenerované, aby výrobce desku plošných spojů dokázal správně vyrobit. Jako výrobce jsme zvolili společnost JLCPCB, přes kterou jsme objednávali desku plošných spojů i na ložskou verzi projektu a poměru rychlosti, kvality a ceny se nám tato společnost osvědčila. Jedna z nevýhod této společnosti je, že sídlí v Číně, takže samotná výroba zde také probíhá. Z tohoto důvodu je nutné počítat s daným časovým zpožděním kvůli dopravě. Toto časové zpoždění lze samozřejmě ovlivnit v závislosti na ceně. Poté co deska přišla přes přepravní společnost DHL z Číny jsme na desku osadili čtyři rezistory a kolíkovou lištu pro následné připojení prototypovací desky plošných spojů, k mikrokontroleru. Poté co jsme provedli test prototypovací desky jsme zjistili, že deska vlastně vůbec nefunguje a nedekuje jenom léky, ale prakticky nic. Proto nás napadlo vzít svítilnu mobilního telefonu a prosvítit pomocí ní prototypovací desku. Po delším zkoumání jsme usoudili, že v druhé vrstvě, pravděpodobně bude chybět samotný senzor, což bylo velice nemilé, protože nás tlačil čas. Mysleli jsme si, že se stala nějaká menší chyba ve výrobě, proto jsme prototypovací desku objednali ještě, jednou a dopsali k ní poznámky, aby nám nepřišli zase desky nefunkční. I když jsme zvolili tu nejrychlejší a nejdražší dopravu, než jsme desku dostali trvalo to něco málo přes týden. Poté co druhá verze naší prototypovací desky přišla z výroby, provedli jsme ten samí pokus se svítilnou. Zjistili jsme, že v druhé vrstvě čtyřvrstvé prototypovací desky opět chybí samotný senzor.



Obrázek 42: Jednotlivé vrstvy prototypovací desky pro kapacitní senzory s červeně vyznačenými částmi, které byli po výrobě zjištěni, že chybí

Tímto problémem jsme se dostali do skluzu už několika týdnů. Schylovala se k tomu, že kapacitní senzory nepoužijeme a vrátíme se zpátky k optické metodě snímání, i když pro nás měl více nevýhod než výhod oproti metodě kapacitní. Po dalším neúspěšném pokusu, jsme se zkontrolovali všechny vrstvy v aplikaci Fusion 360. Vše se zdálo být naprosto v pořádku, ale napadlo nás, že senzor, který byl dělaný jako komponenta, která je uložená v knihovně, může mít při výrobě problém s vrstvami. Protože samotné vrstvy se neupravovali při úpravě samotné desky plošných spojů, ale přímo v komponentě. Takže jsme zkusili tyto čtverce v mědi udělat přímo při úpravě desce plošných spojů. Pak jsme desku znovu odeslali do výroby a doufali jsme, že deska už kapacitní senzor bude mít. Po týdnu přišla třetí verze desek. Znovu jsme je zkusili prosvítit svítilnou a tentokrát zde senzory opravdu byli. Sice jsme ztratila několik týdnů, ale nakonec se povedlo. Po delším testování jsme následně ve finální desce plošných spojů upravili velikost senzorů, tak aby pro nás nejlépe detekovali. Finální desku plošných spojů jsme za nedlouho dodělali, vygenerovali GERBER data a poslali ji do výroby. Paralelně k tomu jsme objednali součástky, bez kterých bychom desky nemohli osadit. Poté, co nám deska a součástky potřebné k osazení dorazili, jsme přešli do fáze osazování součástek. Nejdříve jsme si připravili součástky, které jsme měli v plánu osadit. Pak jsme přilepili k deskám pastovací šablonu (stencil), která přišla z výroby s deskou. Poté jsme na desku plošných spojů museli opatrně nanést pájecí pastu. Po nanášení pasty jsem pinzetou začal desky osazovat. Jelikož jsme používali nejmenší součástky velikosti 0402, desku jsme během několika hodin osadili. Po osazení jsme desku zkontrolovali a následně ji dali na zapájení do pájecí pece na SMD součástky. Po zapájení jsme opravili chyby, které vznikly při zapájení v peci, a ručně dopájeli pomocí mikro pájky chybějící konektor baterie, konektor na připojení USB – C, uživatelské tlačítko a kolíkové lišty pro připojení programátoru a testování. A poté jsme se pokusili desku oživit. Při ožívování projektu vzhledem k velikosti desky plošných spojů jsme zjistili, že došlo k chybám. Jedna z chyb, kterou zmíním bylo například prohození signálových vodičů na sběrnici UART. Tuto sběrnici jsme měli také připojenou na logickém analyzátoru k zjišťování a analyzování problémů na sběrnici. Chyby jsme následně opravili. Jelikož například chybu prohození signálových vodičů na sběrnici UART nebylo možné jinak odstranit než přerušením vodičů na desce a následně jejich prohození pomocí drátků nad deskou, čímž jsme desku poškodili, byla následně objednaná druhá a finální verze desky plošných spojů i s opravenými chybami.



Obrázek 43: Screen z logického analyzátoru při analyzování sběrnice UART

9 NÁVRH KRABICE

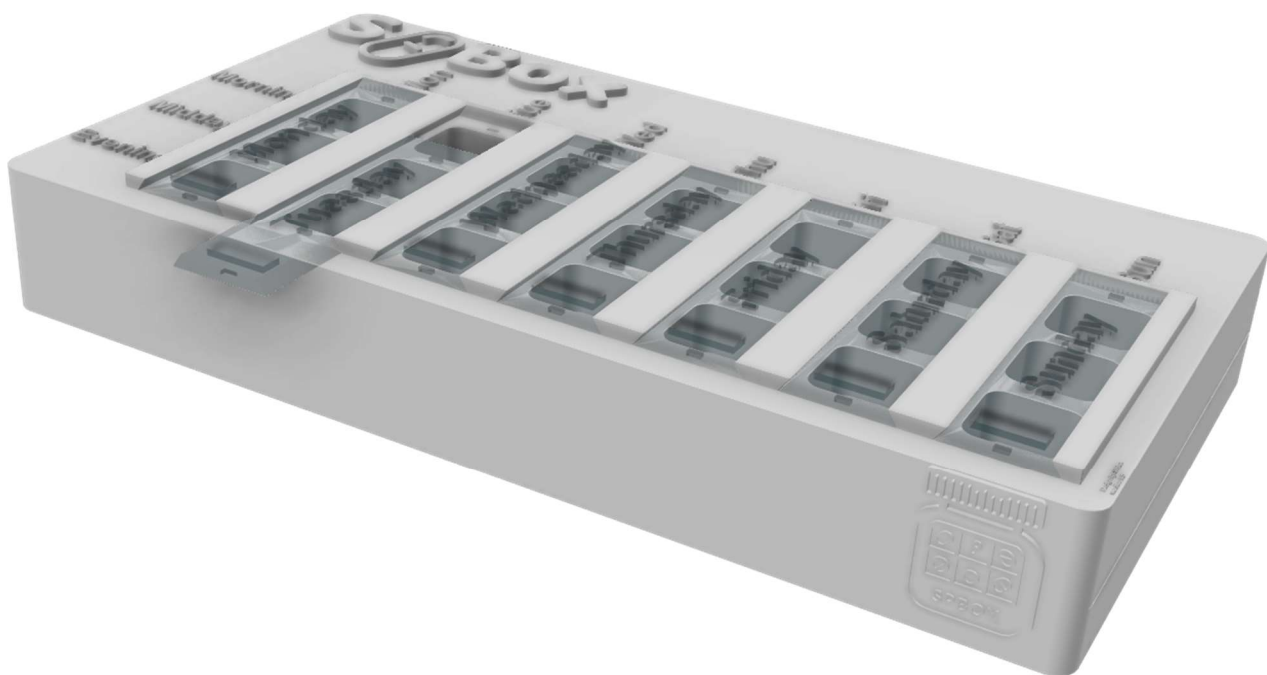
Krabička se celkově skládá ze tří jednotlivých částí a 7 posuvných krytek k zamezení vypadávání prášků.

První částí je dno krabičky, které je vytištěno z bílého filamentu na 3D tiskárně. Je speciálně vytvarované, aby do něj mohla zapadnout deska plošných spojů společně s horní částí krabičky. Ze strany je taky udělán speciální otvor pro uložení akumulátoru a místo na zasunutí krytky. Ta je zde z důvodu, aby se baterie nemohla při pohybu krabičky pohybovat. Jsou zde také čtyři díry pro šrouby M3, pro celkové sešroubování horního a spodní části krabičky. Zespoda jsou v něm čtyři otvory na šrouby M3 sloužící k sešroubování horní a spodní části krabičky.

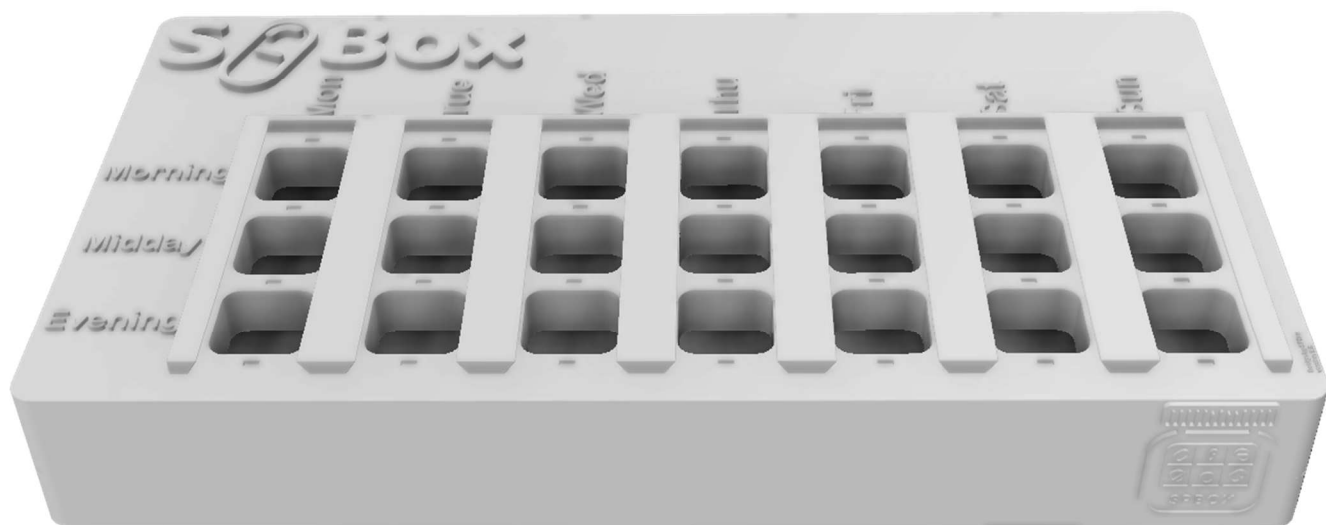
Druhou částí je samotný vršek krabičky vytištěné také bílým filamentem, na kterém jsou na první pohled vidět vyhloubené otvory pro léky, kterých je přesně 21. Nad každým ze samotných otvorů, jsou udělané pouzdra na umístění magnetů, jeden navíc je také vždycky pod posledním z nich, jsou zde z toho důvodu, aby bylo zamezeno tomu, že posuvné krytky budou samovolně vyjíždět z krabičky. Na levé straně je dále možno vidět nápisy, jejichž význam je časově oddělit den na tři části. Na horní části je dále také vymodelované pojezdové kolejnice pro posuvné krytky, které jsou umístěny vedle otvorů na léky. Dále jsou na vrchní části vymodelovány nápisy jednotlivých dnů v týdnu. Na krabičce jsou také umístěna loga projektu. Z levé strany jsou vymodelovány díry, pro uživatelské tlačítko, přepínač pro vypnutí a zapnutí zařízení a také otvor pro samotný USB – C konektor. K horní části je také přišroubovaná samotná deska plošných spojů, ta je přišroubovaná k horní části krabičky, z důvodu zajištění co nejmenší vzduchové mezery mezi samotnou deskou plošných spojů a dnech jednotlivých otvorů na prášky. Je to protože vzduchová mezera by mohla negativně ovlivnit funkci jednotlivých kapacitních senzorů a prášky by tak mohli být detekovány chybně. Deska je přikotvena pomocí sedmi šroubů M3 k závitovým vložkám, které jsou vloženy do distančních sloupků, které byly vymodelovány společně z krabičkou a jsou tak její součástí. Dále jsou zde další čtyři díry, taktéž pro závitové vložky, do kterých se také šroubují šrouby M3. Ty drží vrchní i spodní části krabičky při sobě.

Třetí z částí je již zmíněná krytka na baterii, která je zde k zamezení jejímu pohybu či případnému následnému poškození samotné baterie. Je vytištěná z oranžového filamentu. A je na ní vymodelována ikona baterie, pro jednoznačné rozpoznání místa uložení baterie.

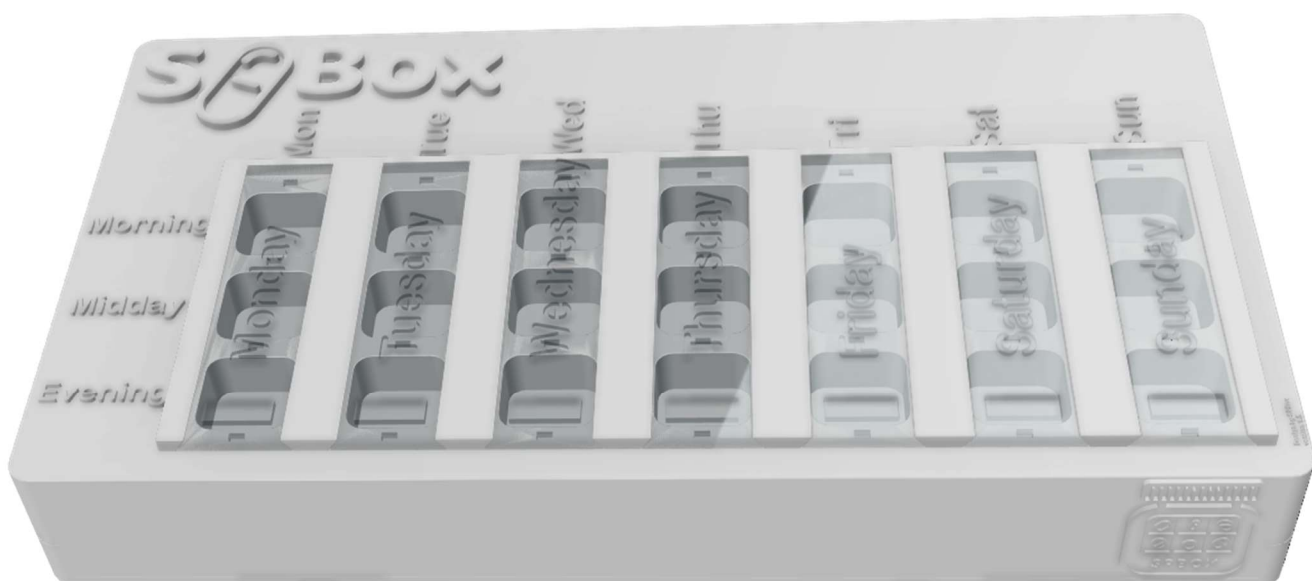
Posuvné krytky, jsou taktéž vytištěny na 3D tiskárně, ale jsou vytištěny z transparentního filamentu, aby bylo pro uživatele co nejjednodušší rozpoznat, jaký z otvorů na prášky je právě podsvícený. A zašoupávají se pomocí kolejníc do horní části krabičky. Zde jsou kotveny pomocí magnetů, které jsou jak na horní části krabičky, tak v samotných posuvných krytkách a navzájem se magnetickými silami přitahují. Takže posuvné krytky by neměli samotně vypadávat.



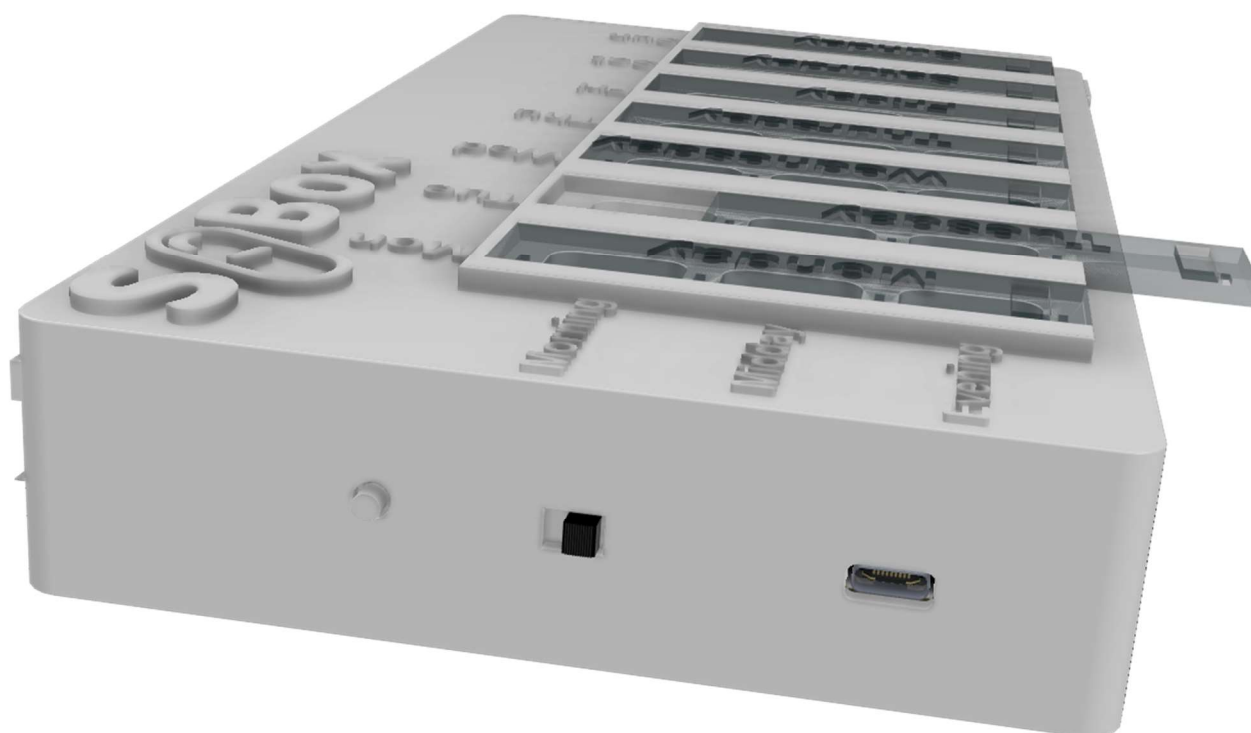
Obrázek 44: Vizualizace modelované krabičky pohled shora



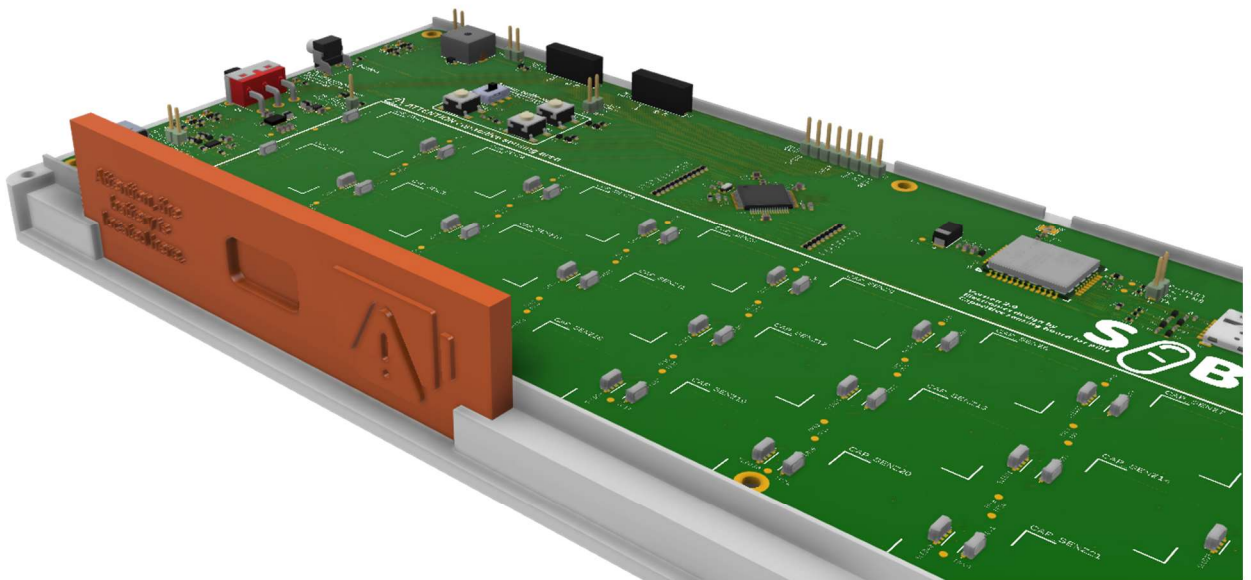
Obrázek 45: : Vizualizace modelované krabičky s šuplíčky



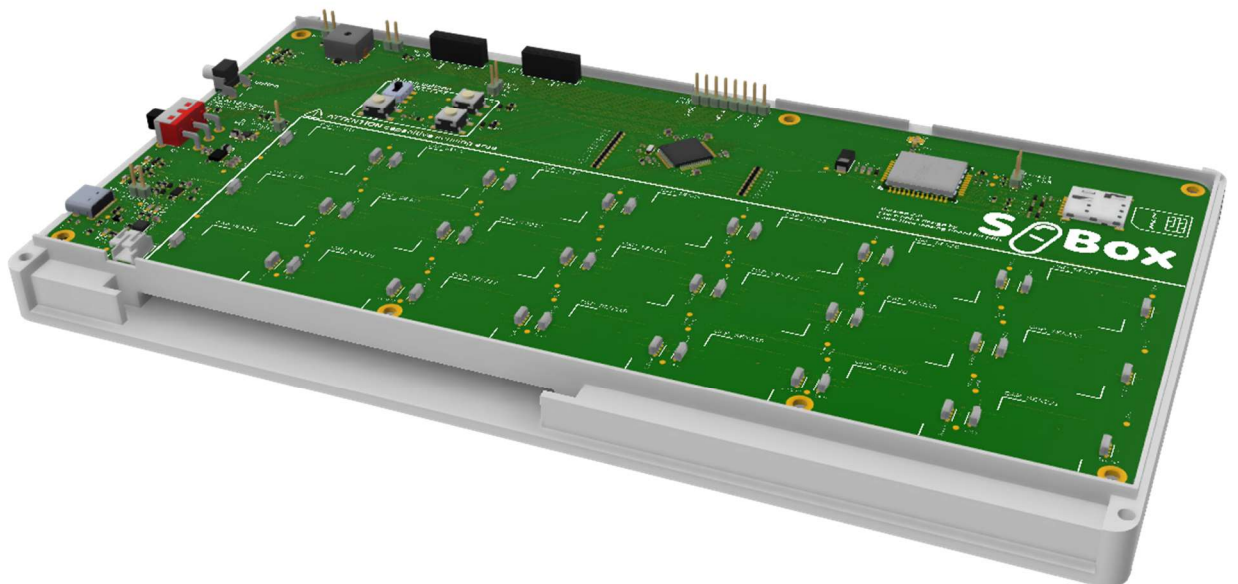
Obrázek 46: : Vizualizace modelované krabičky shora se šuplíčky



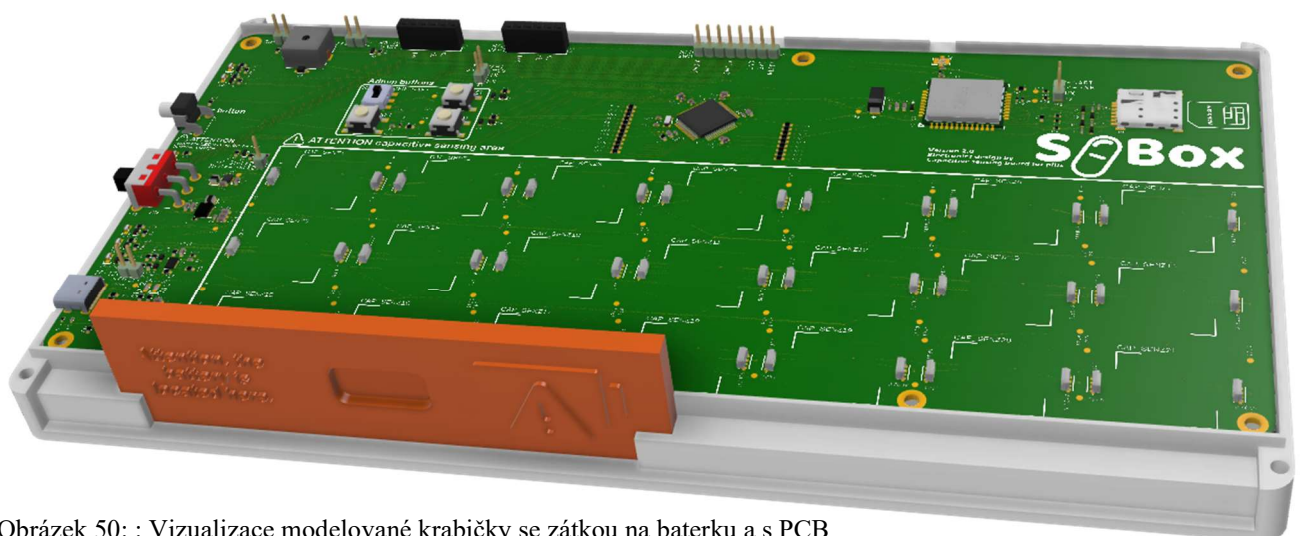
Obrázek 47: : Vizualizace modelované krabičky ze strany



Obrázek 48: Vizualizace modelované krabičky se zátkou na baterku a s PCB



Obrázek 49: Vizualizace modelované krabičky s PCB



Obrázek 50: : Vizualizace modelované krabičky se zátkou na baterku a s PCB

10 NÁVRH LOGA PROJEKTU

Základním nápadem loga byla jednoduchost a jasnost. Rozhodli jsem se do loga použít název a přemýšleli jsme, jakým způsobem ho změnit, aby v něm bylo obsaženo něco, co dává jasnou informaci, o jaký produkt výrobek se jedná. Nakonec nás napadlo použít písmeno P, které v názvu znamená „pill“ (česky prášek nebo lék) a upravit ho do tvaru prášku. První verze první generace loga měla prášek v pozici jako zbytek textu, naproti tomu v druhé verzi jsme pro odlišnost písmena P ve tvaru prášku samotné písmeno natočili.



Obrázek 51: Logo první generace verze 1



Obrázek 52: Logo první generace verze 2

Druhá generace loga projektu, měla taktéž dvě verze a původní nápad a smysl loga se nezměnil. Změnili se jen použité barvy a tvary. Logo bylo celkově zjednodušeno a byl zvolen zaoblenější typ písma. Také byli změněny barvy, byla vyměněna barva modrá za barvu zelenou. Zelená barva by měla na lidi působit klidnějším dojmem a také by měla symbolizovat zdraví a bezpečí. Také byly vytvořeny dvě verze v první verzi má samotný prášek hrany ostré ve druhé verzi je má už zaoblené.



Obrázek 53: Logo druhé generace verze 1



Obrázek 54: Logo druhé generace verze 2



Obrázek 55: Ikona projektu k logu

11 POUŽITÉ KOMPONENTY Z POHLEDU SOFTWARE

Pro řešení vytyčeného cíle jsme se snažili vybrat co nejvhodnější komponenty, které by přispěly k co nejlepšímu a nejefektivnějšímu dosažení cíle. Základním parametrem, na kterém závisel výběr komponent, byla spotřeba. Jelikož se jedná o bateriově napájené zařízení, tak je dosažení co nejvyšší životnosti na jedno nabití jasným cílem.

11.1 PIC32CM5164JH01064

Mikrokontrolery PIC32CM JH00/JH01 jsou postaveny na 32bitové architektuře procesorů Arm® Cortex®-M0+. Jedná se o architekturu 32bitových procesorových jader licencovaných společností ARM Limited. Tato jádra jsou optimalizována pro levné a energeticky efektivní integrované obvody jako jsou MCU. Ty mají velmi nízkou spotřebu. MCU s touto architekturou mají RAM 8–96 kB a flash 64–512 kB. Podporují spánkové režimy, ve kterých má spotřebu nižší než 1 μ A. Tato jádra jsou založena na Von Neumannově architektuře sběrnice.¹⁷ To znamená, že obsahuje pouze jednu paměť pro instrukce a data. To způsobuje nemožnost spravovat programová a instrukční data zároveň.¹⁸

Tyto mikrokontrolery jsou vyráběny s 32 až 100 piny, flash pamětí až 256 kB a SRAM až 40 kB. MCU PIC32CM JH pracuje na frekvenci až do 48 MHz. Obsahuje velké množství periférií jako například: 12kanálový DMA kontrolér, 12kanálový systém událostí, programovatelný kontrolér přerušení, až 84 programovatelných I/O pinů, 32bitové hodiny reálného času a kalendář, až 8 16bitových TC (mohou být nastaveny jako 8bitové/16bitové, vybrané i jako 32bitové), 3 TCC (2 mohou operovat ve 24bitovém režimu a třetí v 16bitovém), až 8 modulů sériové komunikace (každý může být nastaven jako USART, SPI a I²C až do rychlostí 3,4 MHz, dva 12bitové 12kanálové ADC a dvou pinové programovací rozhraní SWD.

Všechna zařízení mají přesné externí a interní oscilátory s nízkou spotřebou. Všechny mohou být použity jako zdroj systémových hodin. Různé hodinové domény mohou být nezávisle nakonfigurovány, aby běžely na různých frekvencích. Mohou být nastaveny podle periférií, aby běžely na optimální frekvenci.

PIC32CM JH má 2 softwarově nastavitelné režimy spánkového režimu: Nečinný a Pohotovostní. V nečinném režimu je MCU zastaveno, ale všechny ostatní funkce zůstávají v provozu. V pohotovostním režimu jsou všechny hodiny a funkce zastaveny vyjma těch, které jsou nastaveny, aby byly v provozu. V tomto režimu je paměť a logika zmrazena.¹⁹

¹⁷ (Arm Ltd., 2020)

¹⁸ (Kutý, Nedatováno)

¹⁹ (Microchip Technology Inc., 2022)

11.2 FreeRTOS

Firmware na tomto MCU je psán v operačním systému FreeRTOS. Jedná se o volně šiřitelný operační systém reálného času (RTOS), který běží na mnoho mikrokontrolerech. V roce 2017 nad ním převzal kontrolu Amazon a dnes provádí jeho pravidelnou údržbu a podporu. Každý výrobce mikrokontrolerů a mikroprocesorů poskytuje vlastní upravenou verzi postavenou čistě na své produkty.

RTOS je operační systém, který spravuje běh úloh a zajišťuje jejich rovnoměrné vykonávání. Proto se někdy může hovořit o téměř souběžném fungování. Je vhodný pro aplikace, ve kterých je klíčové dodržet přísné časové lhůty. Průběh je založen na ticích, každý tik MCU je zvolena úloha, která bude do dalšího tiku probíhat. Každá úloha má své jméno, název funkce, která se volá, velikost zásobníku úlohy, vstupní parametry, prioritu a obsluhovač. Jméno je text, který slouží pouze jako označení, v kódu není nijak použité. Vstupní parametry mohou, ale nemusí být uvedeny, podle toho, zda funkce nějaké vstupní parametry vyžaduje. Priorita značí důležitost úlohy. Úlohy s vyšší prioritou mají větší pravděpodobnost, že budou vykonány. Obsluhovač je určen pro ovládání úlohy (například zrušení).

Pokud je připraveno být vykonáno více úloh se stejnou prioritou, je spuštěná úloha volena náhodně. Všechny úlohy, ale musí mít možnost být vykonány tak, aby nebyly zpožděny. Jakmile je připravena ke spuštění úloha s vyšší prioritou, bude vybrána právě ona. Bude volána do doby, než nebude dokončena. Ostatní úlohy musí na její dokončení čekat. Nejvyšší prioritu mají externí přerušeni, které jsou oprávněny přerušit úlohu s jakoukoli prioritou i během jejího vykonávání. Po dokončení obsluhy přerušeni pokračuje úloha v provozu. Úlohy mohou být ve čtyřech stavech. Prvním stavem je stav „Running.“ V tomto stavu se úloha vykonává. Druhým stavem je „Ready.“ Úloha v tomto stavu je připravena být vykonána, ale není vykonávána. Čeká na dokončení jiné úlohy stejné nebo vyšší priority. Třetím stavem je stav „Blocked.“ Úlohy v tomto stavu čekají na časovou nebo vnější událost. Například pokud funkce zavolá příkaz pro systémový časovač, nebo čeká na řadu, mutex nebo semafor. Posledním režimem je režim „Suspended.“ Tento režim je podobný jako režim „Blocked“ s tou výjimkou, že mohou být z tohoto nebo do tohoto režimu přesunuty pouze vykonáním funkce.

Pro práci s úlohami, jejich synchronizaci a sdílení dat jsou v RTOSu definovány tyto hlavní funkce: fronta, notifikace, mutex, semafor a notifikace. Fronta je funkce pro předávání dat z jedné úlohy do druhé. Řeší problém používání globální proměnné. Pokud je globální proměnná používána jednou úlohou, která čte, nebo zapisuje data a je spuštěna jiná úloha, která také s touto proměnnou pracuje, může dojít k přepsání části nebo celých dat. Řada je definována s nějakou délkou a velikostí jednoho objektu v ní uloženého. Funguje na principu první dovnitř, první ven. To znamená, že data, která byla do řady uložena nejdříve budou také nejdříve vyčtena. Při vyčítání jsou vždy vyčtena nejstarší data. Úloha může být zablokována ve dvou případech. Prvním případem je, když se úloha snaží do řady zapsat data, ale řada je plná, v tomto případě bude úloha odblokována po uvolnění místa v řadě. Druhou možností je

zablokování úlohy v případě, že se snaží vyčíst data z řady, ale řada je prázdná. Je odblokována po naplnění řady.

Mutex je možnost, jak využívat globální proměnnou a mít jistotu, že data v ní během jejího využívání jiná úloha nepřepíše. Slouží jako „klíč“ k datům. Funguje tak, že pokud úloha chce pracovat s globální proměnnou, nejdříve zkontroluje, zda je mutex plný. Pokud je plný, vezme si klíč a začne pracovat s proměnnou. Po dokončení opět klíč vrátí. Jakmile by jiná úloha chtěla pracovat s tou samou proměnnou, tak musí nejdříve zjistit, zda je plný, ale když zjistí, že je prázdný, přesune se do blokováného režimu, než je mutex znovu naplněn.

Semafor slouží k upozornění, že jsou k dispozici nějaká data, nebo že proběhla nějaká událost. Existují dva typy semaforů. Prvním typem je binární semafor, tento semafor má pouze dva stavy. Ty jsou využívány k jednoduché indikaci. Druhým typem je čítací semafor, ten má definovaný počet stavů. Slouží pro indikaci většího množství stejných dat. Výchozí stav semaforu je 0. Pokud chceme, aby nějaká úloha něco vykonala, nastavíme semafor. Když úloha semafor vyčte, změní se hodnota semaforu na 0. Dá se použít například při obsluze přerušení. Nastavování čítacího semaforu nebo vyčtení funguje na stejném principu jako u fronty. Méně náročnou metodou upozornění úlohy je notifikace.

Pro časování jednotlivých úloh je možné místo hardwarového časovače použít softwarový časovač. Pro jeho vytvoření ho musíme pojmenovat, nastavit mu periodu, nastavit režim, identifikační číslo a funkci pro zpětné volání. Perioda časovače je udávána v ticích, proto je nutné požadovanou hodnotu vydělit periodou tiku. Režim časovače může být buď jednorázový, nebo periodický. Výhodou softwarového časovače je možnost vytvoření velkého množství časovačů, nevýhodou je menší rozlišení.

Hardwarová přerušení jsou přerušení vyvolána externí periferií. Po přerušení je vždy volána jeho obslužná funkce. Tato funkce má vždy vyšší prioritu než úlohy. Je schopna přerušit právě běžící úlohu a běžet místo ní do doby, než bude dokončena. Z toho důvodu by měla být tato funkce co nejkratší a nemělo by v ní docházet ke dlouhým procesům. Často jsou v něm volány funkce například pro nastavení semaforu. Nesmí být ale volány obyčejné funkce, ale ty určené pro vykonání při přerušení. Názvy těchto funkcí končí slovy „FromISR.“²⁰

RTOS potřebuje RAM vždy když je založena nová úloha, mutex, softwarový časovač atd. Paměť může být automaticky dynamicky alokována z haldy RTOSu nebo může být poskytnuta programátorem. RTOS má několik implementací alokace paměti ve zdrojovém kódu. Heap_1 je nejjednodušší a nepodporuje uvolnění paměti. Je využitelná, pokud nepotřebujeme jednotlivé úlohy mazat. Heap_2 podporuje uvolnění paměti, ale nespojuje sousední volné bloky. Heap_3 dělá funkce „malloc()“ a „free()“ bezpečnými pro vlákna. Heap_4 spojuje sousední volné bloky

²⁰ (ShawnHymel, Nedatováno)

pro vyhnutí se fragmentace, obsahuje možnost umístění absolutní adresy. Heap_5 je stejná jako heap_4 se schopností rozprostřít haldu do více nesousedících paměťových oblastí.²¹

11.3 Kapacitní snímání

Následující informace jsou citovány z aplikační zprávy pojednávající o kapacitním snímání: „Kapacitní snímání začíná být velmi populární technologií nahrazující optickou detekční metodu a mechanické návrhy pro aplikace jako detekce přiblížení nebo gesta, materiálů a snímání hladiny kapaliny. Hlavními výhodami, které má kapacitní senzor oproti jiným způsobům detekce, jsou, že je schopen snímat mnoho rozdílných materiálů jako pokožku, plast, železo nebo kapalinu, snímá bezkontaktně a neopotřebovává se. Je schopen detekovat až do velkých vzdáleností s malou velikostí senzoru samotného, senzor vytvořený na desce plošných spojů je velice levný a jeho funkce je s nízkou spotřebou.“²²Tento úryvek byl citován, jelikož dokazuje vhodnost použití kapacitní technologie pro náš účel.

Použitý mikrokontroler obsahuje periférii PTC, která je vyvinuta společností Microchip a je zodpovědná za kapacitní měření. Informace uvedené v následujících odstavcích jsou čerpány z uživatelské příručky o jejím užití. Senzor je vytvořen na desce plošných spojů a připojen na pin, jenž jí náleží. Podporuje dva typy kapacitního měření, režim vlastní a vzájemné kapacity. Sensory pracující na principu vlastní kapacity měří náboj mezi senzorovou deskou a plochou s nulovým potenciálem, při vzájemné náboj mezi dvěma elektrodami senzoru (vysílač a přijímač).

Pro kapacitní měření se ve vývojovém prostředí MPLAB X IDE od společnosti Microchip v zásuvném modulu MCC využívá QTouch Library (tedy knihovna pro ovládání periferie kapacitního snímání tedy PTC) a PTC periferie. Ta podporuje velkou citlivost, toleranci šumu a automatickou kalibraci minimalizující nastavování citlivosti uživatelem. Každý senzor je připojen na jeden kanál, pomocí, kterých se identifikují. Samotná knihovna se dá rozdělit na tři hlavní moduly. Modul pro komunikaci s periferií, který obstarává měření, řazení kanálů, kalibraci a další parametry. Poskytuje 3 hlavní data, a to hodnotu signálu, reference a delty. První parametr obsahuje syrová naměřená data na aktuálním kanálu. Reference udává průměrnou hodnotu signálu zaznamenávanou po delší dobu a třetí je rozdíl mezi hodnotou signálu a reference. Další modul obstarává změnu parametrů signálu jako přepínání budicí frekvence, filtrování a měření šumu a poslední provádí zpracování obdrženého naměřeného signálu například vyhodnocování stisknutí a pozice prstu.²³

²¹ (Amazon Web Services Inc., Nedatováno)

²² (Texas Instruments Inc., 2021)

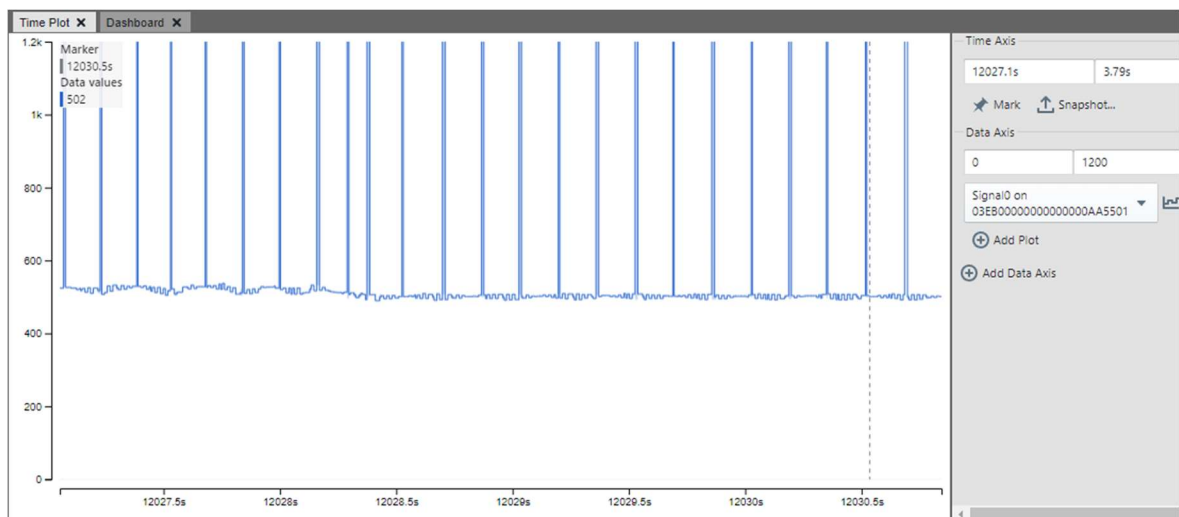
²³ (Microchip Technology Inc., 2020)

11.3.1 Testování PTC periferie

Prováděli jsme testování jednotlivých typů kapacitních senzorů. Nejdříve jsme otestovali kapacitní senzory umístěné na výukových deskách k MCU SAML21 a PIC32CM JH. Tyto senzory byly typu měření vlastní kapacity. Zkoušeli jsme, jestli při položení políčka a vložení léku, bude lék detekovatelný. Zjistili jsme, že je změna kapacity detekovatelná. Rozhodli jsme se otestovat druhý typ senzorů, který funguje na principu vzájemné kapacity. Tyto senzory jsme otestovali a zjistili jsme, že více kapacitu ovlivní ruka položená nad políčkem, než samotný lék. Poté jsme se rozhodli vyrobit si vlastní senzory, které mají stejný design jako senzory na výukových deskách. Vyzkoušeli jsme několik velikostí a zjistili, že by pomocí nich detekce mohla fungovat.

Přes zásuvný modul programu MPLAB X IDE MCC v nastavení „Touch“ je možné nastavit typ kapacitního senzoru (vlastní a vzájemná kapacita), jestli bude pracovat jako senzor s nízkou spotřebou. Poté je pro každý senzor nastaven pin, digitální a analogový zisk, stupeň filtrace, práh, hodiny a hystereze. Dále je možné nastavit aktivní stínění a další.

Měření probíhá po zavolání funkce `touch_process()`. Pokud je nastavený příznak indikující, že



Obrázek 56: Testování kapacitního senzoru pomocí zásuvného modulu data visualizer

je čas na měření kapacity, zahájí se měření. Po skončení měření se pomocí funkce zpětného volání zavolá zpracování dat. Po úspěšném zpracování je nastaven příznak `measurement_done_touch`. Pro odladování parametrů a nastavení senzoru je k dispozici funkce `touch_data_streamer`, která posílá přes UART data jako signál, reference, stav, práh a další. Tato data je poté možno zobrazit přes zásuvný modul MPLAB Data Visualizer. Pro získání naměřených dat je možné zavolat funkce:

- `get_sensor_node_signal()`
- `get_sensor_node_reference()`
- `get_sensor_cc_val()`
- `get_sensor_state()`.

11.4 BC660K

BC660 je vysoce výkonný vícepásmový modul LTE Cat NB2 s extrémně nízkou spotřebou. Podporuje mnoho protokolů a poskytuje bohaté externí rozhraní. Má vyhrazenou zabudovanou eSIM. Podporuje protokoly jako UDP, TCP, MQTT, CoAP, HTTP. Aktualizaci firmware a nastavení modulu lze provést přes UART nebo DFOTA. Ve vývoji jsou i SMS, které může podporovat textový nebo PDU režim. Komunikace s modulem je řešena formou AT příkazů. Pro snížení spotřeby energie zařízení podporuje spánkové režimy. Prvním režimem je PSM (Power Saving Mode) a druhým eDRX. Tyto dva režimy je možno kombinovat.²⁴

11.4.1 Komunikační technologie NB-IoT

Narrow band (NB-IoT) je stejně jako jiné 3GPP technologie poskytován v licencovaném pásmu a může být používán pomocí modulů. Technologie je poskytována mobilními operátory poskytujícími LTE sítě. Uživatelé se mohou připojit k síti pomocí APN poskytovatelů pomocí SIM. Každý operátor má své veřejné APN, které jsou specifické pro jejich lokální nabídky. Uživatelé se mohou také připojit na své privátní APN. NB-IoT poskytuje několik možností, jak snížit spotřebu.

První možností je Long-Periodic Tracking Area Updates (Long-Periodic TAU). Tato funkce je užita pro periodické upozornění síti o dostupnosti zařízení. Tato procedura je kontrolována modulem pomocí periodického časovače (T3412). Výhodou Long-Periodic TAU je, že modul může zůstat v hlubokém spánkovém režimu po delší čas, před tím, než se musí probudit, aby poslal TAU zprávu. Periodu časovače je možné nakonfigurovat.

Další možností, jak dále snížit spotřebu je Power-Saving Mode (PSM). Tato funkce pomáhá zařízením zachovat stav baterie a prodloužit její životnost. Ačkoli je možné vypnout rádio pro prodloužení životnosti baterie, po zapnutí je nutné se znovu připojit k síti. Tato činnost spotřebovává značné množství energie. Pro toto je k dispozici právě alternativa v podobě PSM, který deaktivuje části modulu, a tak sníží spotřebu do řádů mikroampérů, zatímco zůstává připojen k síti. Pro funkci PSM jsou potřebné dva časovače: PSM Activity Timer (T3324) a Long-Periodic TAU Timer (Rozšířený T3412). PSM Activity Timer určuje čas, během kterého zařízení zůstává v režimu nečinnosti a naslouchá paging zprávám. Long-Periodic TAU časovač definuje čas mezi dvěma TAU. Během tohoto času modul zůstává v hlubokém spánkovém režimu. Během tohoto času si síť zachovává informace o stavu zařízení a modul zůstává registrovaný. Po probuzení se modul nemusí znovu připojit a ušetří energii.

Třetí možností ke snížení spotřeby zařízení je funkce Extended Discontinuous Reception (eDRX). Tato funkce je rozšíření existující LTE funkce, která je užívána pro větší snížení spotřeby IoT zařízení. Je designována pro zařízení, která častěji data přijímají, než odesílají a

²⁴ (Quectel Wireless Solutions Co., 2023)

je používána hlavně v zařízeních, kdy není kritické, když je zařízení nedosažitelné po dobu od několika sekund do pár hodin. Je určena pro aplikace, kde je potřebné, aby se zařízení probouzelo a v periodických intervalech a přijímalo příchozí data. eDRX má časový interval, během kterého zařízení neposlouchá síť. Tento interval může být zvýšen pro zvětšení úspory energie. Během tohoto intervalu je zařízení stále dostupné v síti. Funkce eDRX může být použita bez PSM, nebo společně s PSM pro větší snížení energie. eDRX neposkytuje stejnou úsporu energie jako PSM, ale poskytuje dobrý kompromis mezi dostupností a spotřebou. Obsahuje dva typy časovačů: Paging Transmission Window (TPTW) a eDRX Cycle (TeDRX). První zmíněný časovač udává čas, během kterého zařízení probíhá eDRX procedura. Délka tohoto intervalu může být mezi 4 a 16 paging přijímacími sloty. Druhý zmíněný časovač udává čas mezi začátky dvou po sobě jdoucích PTW. Mezi těmito dvěma časy modul vstoupí do spánkového režimu, během kterého je příjem dat deaktivován. Hodnotu časovačů je možné nastavit v poskytovaných rozsazích.

Poslední funkcí pro snížení spotřeby je Release Assistance Indication. Toho je dosaženo tím, že IoT zařízení může předčasně přerušit nosič RRC na 3. vrstvě mezi sebou a eNodeB v rádiové přístupové síti operátora. To je umožněno přidáním Release Assistance Indicator IE během odesílání dat, pro informování sítě, že se neočekává žádný uplink ani downlink přenos. Aktivováním Early Release Indication je zařízení umožněno přejít přímo do nečinného režimu po odeslání nebo příjmu dat. Díky tomu dojde k přeskočení stavu RRC_CONNECTED. V závislosti na modulu může být ušetřeno až 50 mA.

Nevýhoda použití PSM a eDRX je, že IoT zařízení nemohou být kontaktovány sítí, když je zařízení ve spánkovém režimu. Nicméně během tohoto času jsou data poslaná zařízení uložena sítí a odeslána po probuzení. Tato funkce se nazývá „Data Packet Buffering.“ Počet a velikost datových paketů, které jsou uloženy je závislá na použité síti.

NB-IoT má velmi dobré pokrytí i v prostorech, které nejsou dosažitelné 2G, 3G a LTE sítěmi, jako například podzemí. Toho je dosaženo opakováním zpráv 3. vrstvy. NB-IoT standart podporuje tři takzvané Coverage Enhancement (CE) Levels. Každý level definuje počet opakování downlink a uplink zpráv, pro komunikaci se zařízením v oblasti se špatným signálem. CE Level 0 je používán v prostorech, kde je dobré pokrytí. CE Level 1 je používán v prostorech se ztrátou +10 dB. Počet opakování je střední. V prostorech se ztrátou až + 20 dB je používán CE Level 2. Počet opakování je až 128. Při CE Level 1 a 2 je také použitý vyšší vysílací výkon.

Síť NB-IoT je možné díky roamingu i v zahraničí díky spolupráci poskytovatelů sítě a roamingovými partnery. Minimální funkce, které jsou zajištěny ve všech sítích roamingových partnerů jsou PSM a Long-Periodic TAU.²⁵

²⁵ (Deutsche Telekom Inc., 2023)

11.4.2 Testování komunikace s modulem pomocí konzole

Pro komunikaci s komunikačním modulem BC660 se používají AT příkazy. Testoval jsem ji pomocí výukové desky s modulem BC660. Na desce je převodník USB na UART. Komunikaci jsem prováděl pomocí sériového terminálu YAT. Výhodou tohoto terminálu je možnost uložit AT příkazy a nemuset je neustále psát. Nejdříve si uživatel zvolí typ terminálu (textový/binární) a jestli se jedná o sériový terminál, nebo internetový. V případě, že se jedná o sériovou komunikaci, zvolí se COM port, počet znaků za sekundu, počet datových bitů, parita a stop bit. Poté již uživatel již může přímo komunikovat, nebo si vytvořit příkazy a uložit je.

Základní nastavení modulu probíhá pomocí příkazů v následujících krocích:

1. AT+I – vypsání identifikačních informací o modulu
2. AT+QCCID – vyčtení identifikačního čísla SIM karty, ta musí být vložena a aktivována
3. AT+QCFG? – vyčtení nastavení modulu
4. AT+CFUN=0 – vypnutí funkcionality modulu
5. AT+QCFG="relversion",14 – nastavení verze vydání 3GPP NB-IoT 14
6. AT+QCFG="NBcategory",2 – nastavení verze na NB2
7. AT+CFUN=1 – zapnutí funkcionality modulu
8. AT+QCGDEFCONT= "IPV4V6","nb.m2mc" – nastavení jména přístupového bodu
9. AT+CEREG=5 – nastavení množství obdržovaných nevyžádaných zpráv o síti
10. AT+CEREG? – výpis stavu připojení k síti
11. AT+CGPADDR? – výpis přidělené IP adresy
12. AT+QIOPEN=0,0,"UDP","192.168.0.20",4242,5000 – otevření komunikačního kanálu
13. AT+QISEND=0,5,"12345" – odeslání dat

Pro lepší příjem dat jsem se rozhodl vypnout PSM což zlepšilo dostupnost modulu. To je možné provést příkazem „AT+CPSMS=0.“ Nastavením na 1 se PSM zapne. Modul lze resetovat pomocí příkazu „AT+QRST=1.“

V platformě Miotiq, která slouží jako síťový server pro přeposílání dat na aplikační, lze zprávy ze zařízení přeměrovat buď pomocí UDP protokolu, nebo pomocí HTTPS. Zprávy z aplikačního serveru lze přeměrovat pomocí získání dat z HTTPS protokolu a přeposlání přes UDP do zařízení. Pro tuto aplikaci byl použit HTTPS protokol. Na aplikačním serveru byl

```
+IP: 10.0.21.213

+CEREG: 5,"FDE8","0C61BBAC",9,,,"11100000","01011111"
AT+QIOPEN=0,0,"UDP","192.168.0.20",4242,5000
AT+QIOPEN=0,0,"UDP","192.168.0.20",4242,5000
AT+QIOPEN=0,0,"UDP","192.168.0.20",4242,5000<CR>
OK

+QIOPEN: 0,0
AT+QISEND=0,5,"12345"
AT+QISEND=0,5,"12345"
AT+QISEND=0,5,"12345"<CR>
OK

SEND OK
AT+CPSMS=0
AT+CPSMS=0
AT+CPSMS=0
AT+CPSMS=0<CR>
OK

+QIURC: "recv",0,5,"12345"
+QIURC: "recv",0,5,"12345"
```

Obrázek 57: Komunikace s modulem BC660

```
28. 11. 2023 15:31:55 node: debug 19
msg.payload : Object
  ▶ { customerId: "02150107", rcvTime:
    1701181915, srcImsi: "901288002250958",
    srcIP: "10.0.21.213", srcPort: "5000" ...
  }

28. 11. 2023 15:31:55 node: debug 19
msg.payload : string[5]
"12345"

28. 11. 2023 15:32:09 node: debug 20
msg.payload : string[2]
"OK"

28. 11. 2023 15:32:09 node: debug 20
msg.payload : string[2]
"OK"
```

Obrázek 58: Node-RED výstup ladicího nodu

vytvořen pomocí vstupního HTTP nodu a nodu HTTP odpovědi tok pro zpracování přijatých dat. Jelikož jsou data zakódována v base 64, musí se HTTP data dekodovat. Finální dekodovaná data byla zapojena do nodu lazení. Ve vstupním nodu jsem nastavil metodu POST a URL „/pdata,“ to byla adresa, kam byla v platformě Miotiq data přeměrována. Node HTTP odpovědi neodesílal žádnou odpověď. Poté jsem vytvořil odesílání dat, které jsem sestavil z vkládacího nodu, funkce pro vytvoření HTTP zprávy a zakódování dat do base 64 a nodu pro HTTP požadavek. V nodu HTTP požadavek jsem nastavil metodu post a jako URL jsem nastavil adresu mého API, podle dokumentace platformy.

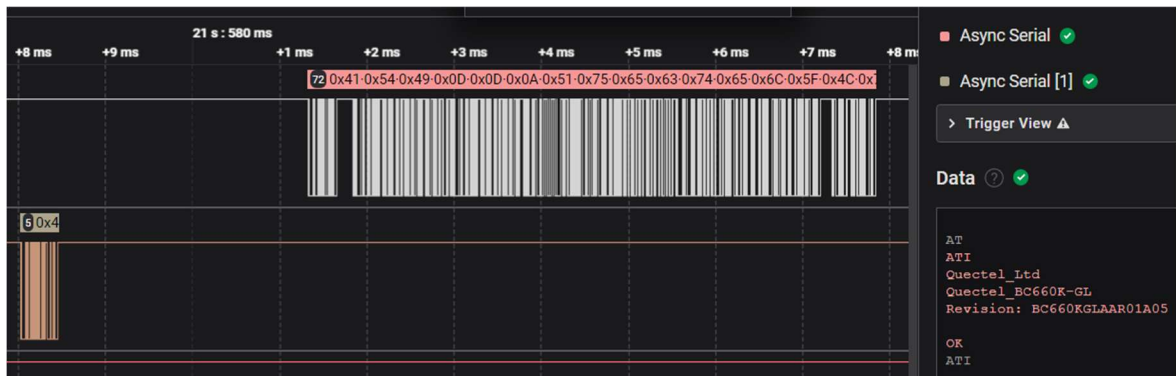
11.4.3 Testování komunikace s modulem pomocí MCU

Pro komunikaci mikrokontroleru se NB-IoT modulem přes sériovou asynchronní sběrnici UART jsem musel vytvořit funkce, které zakódují, nebo dekodují. Modul komunikuje pomocí AT příkazů. Nejdříve jsem si deklaroval proměnné, které obsahují klíčové příkazy. Periferii UART jsem nastavil do režimu kruhové vyrovnávací paměti a nastavil její velikost pro zápis i výčet. Rychlost periferie jsem nastavil na výchozí hodnotu modulu, kterou je 115 200 Hz. Dále jsem si založil pomocné proměnné a pole.

Funkci čtení dat z kruhové vyrovnávací paměti pokaždé, když jsou v ní uložena nová data vyčte 1 byte, který se uloží do pomocné proměnné a následně převede na znak, jenž je uložen do náležitýho indexu pole znaků. Pokud není přečten znak „\n“, index se inkrementuje a celý proces je vykonán znovu. V opačném případě se index vynuluje. Čtení je voláno ve funkci a při kompletním přečtení řetězce je zavolána funkce na vyhodnocení příkazu. Vyhodnocování se provádí pomocí funkce strstr(), která je součástí knihovny <string.h>. Tato funkce slouží

k vyhodnocení, zda jeden řetězec obsahuje druhý řetězec. Funkce na vyhodnocení příchozí zprávy obsahuje řadu podmínek, ve kterých se vyhodnocuje, o jaký typ zprávy se jedná. Zprávy typu „+IP“ a „+CEREG“ jsou nevyžádané zprávy a jejich příchod není na ničem závislý. Pokud je přijata zpráva „+IP“ značí to, že je zařízení úspěšně připojeno k síti. Zpráva „+CEREG“ obsahuje také bližší informace jako stav připojení a nastavení sítě. Dalšími nevyžádanými zprávami jsou zprávy typu „+QIURC“. Takovéto zprávy mohou přijít v případě obdržených dat, nebo v případě zavření socketu vzdálenou stranou. Zpráva s obdrženými daty je ve formátu „+QIURC: "recv",0,4, "1234"“ a se zprávou o zavření socketu „+QIURC: "closed",0.“ V případě přijatých dat jsou zpracována a vyzvednuta z řetězce pomocí funkce `sscanf()`, jež je součástí základní knihovny jazyka C, `<stdio.h>`. Tato funkce slouží k dekonstruování dat z textového řetězce do proměnných. U ostatních zpráv se jedná o odpovědi na požadavky, které obsahují v případě úspěchu „OK“ a v případě chyby „ERROR.“ Výjimkou je zpráva pro otevření socketu, kdy přijde po odpovědi ještě zpráva od sítě např. „+QIOPEN: 0,0.“ Odpovědi na požadavky se skládají z opakování příkazu, odřádkování a reakce. Pokud je zjištěno, že přijatá data jsou stejná jako odeslaná (opakování požadavku), tak jsou v cyklu vyčítána přijatá data až do doby, kdy je obsažena reakce, podle níž je zpráva dále zpracována.

Odesílání dat je o hodně jednodušší než příjem, jelikož odpadá nutnost zpracování a vyhodnocování. Odesílání dat jsem vyřešil pomocí funkce `sprintf()`, která je součástí knihovny `<stdio.h>`. Tato funkce slouží ke zkonstruování textového řetězce z více proměnných. Funguje podobně jako `printf()`, ale výstup ukládá do proměnné. Dále je zavolána funkce pro odeslání dat, ta obsahuje cyklus `for`, jež slouží k převodu textu na číslo, z důvodu, že funkce UART pro



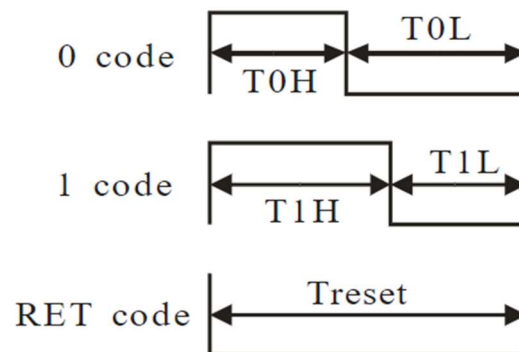
Obrázek 59: Logická analýza – komunikace s NB-IoT modulem BC660

zápis i čtení dat podporuje jako parametr pouze datový typ `uint8_t`. V cyklu je zakomponována jednoduchá podmínka, která ověřuje, zda je aktuální znak „\n.“ Pokud je podmínka splněna, data jsou odeslána a cyklus zrušen. V opačném případě se pokračuje v cyklu.

11.5 Programovatelná RGB LED s integrovaným obvodem WS2815

Jedná se o RGB diodu, jejíž barva se nastavuje pomocí šířky pulzu. Čip rozpoznává 3 typy sekvencí: 0, 1, RESET. Sekvence 0 a 1 se skládají z hodnot napětí 1 a 0. Perioda pro kód 0 se skládá ze signálu HIGH, který trvá pod dobu 220–380 ns a signálu LOW trvajících 580 ns–

Sequence Chart



Obrázek 60: Sekvence čipu WS2815. Získáno z datového listu diody IN-PI42TAS(X)R(X)G(X)B. Inolux Corp. Str. 5. 2021. Dostupné z: [https://www.inolux-corp.com/datasheet/SMDLED/Addressable%20LED/IN-PI42TAS\(X\)R\(X\)G\(X\)B_v1.1.pdf](https://www.inolux-corp.com/datasheet/SMDLED/Addressable%20LED/IN-PI42TAS(X)R(X)G(X)B_v1.1.pdf)

1 μ s. Sekvence pro kód 1 se skládá ze signálu HIGH, jenž zabírá čas 580 ns–1 μ s a LOW, který trvá 220–420 ns. RESET se používá pro resetování čipu (vypnutí LED) a skládá se pouze ze signálu LOW trvajících více než 280 μ s. Těmito sekvencemi se nastaví binární kód odpovídající barvě, kterou LED svítí. Celkem během jednoho cyklu dioda přečte 24 bitů dat. Prvních 8 bitů je nastavení zelené složky, druhých červené a posledních modré. Zbylá data jsou poslána následující diodě. LED přijímá data na vstupu DI (vstup dat) a zbylá data posílá výstupem DO (výstup dat).²⁶

11.6 Testování komunikace s LED

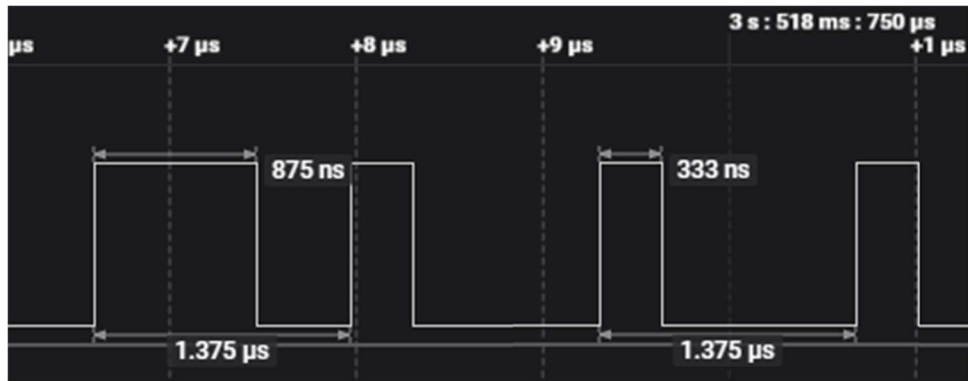
Z důvodu ušetření pinů na mikrokontroleru a většímu přizpůsobení barev jsme se rozhodli použít RGB diody s integrovaným čipem WS2815. Díky tomuto čipu odpadá nutnost mít pro každou LED diodu vlastní signálové cesty. Tyto LED mají pouze čtyři vývody, z toho dva napájecí, jeden datový vstup a jeden výstup. Barvu, kterou dioda svítí, lze nastavit pomocí kódu. Kód pro jednu diodu se skládá ze 24 bitů, z nichž každých 8 bitů náleží jedné barvě. Zbylý kód je poslán dalším diodám na datovém výstupu. Pro nastavení nové barvy je nutné poslat resetovací kód.

Z důvodu, že v MCU není žádná periférie pro komunikaci s čipem, musel jsem ji realizovat softwarově. Jednou z možností by byla synchronní komunikace. Tu by se dalo realizovat

²⁶ (Inolux Corp., 2021)

pomocí nastavení výstupu na určitou logickou úroveň, počkat předem definovanou dobu, podle datového listu čipu a úroveň změnit. To by byla jednoduchá varianta, ale vysoce neefektivní, jelikož by to zablokovalo všechny ostatní činnosti. To by byl problém hlavně, kdyby nebyl použit FreeRTOS, ale klasický stavový automat. I tak jsem se ale rozhodl tuto metodu nepoužít a vytvořit komunikaci asynchronním způsobem. Pro komunikaci jsem se rozhodl použít periférii SPI. Jako signálový vodič fungoval signál MOSI (výstup řídicího zařízení), zbylé signály jsem nechal nezapojené. Pro SPI jsem se rozhodl z toho důvodu, že mají datové vodiče ve výchozím stavu hodnotu HIGH, dosahuje dostatečné rychlosti a jsou posílána samotná data. U sběrnice I²C je problém s rychlostí, adresou, bitem pro indikaci čtení a zápisu a potvrzovacími bity. U UARTu je rychlost příliš malá, ale překážejí i start a stop bity. Frekvenci SPI jsem nastavil tak, aby byla optimální pro signalizaci data pro čip. V kódu jsem si nejdříve definoval pole obsahující tři typy signálů, a to logickou jedničku, logickou nulu a reset (ten je nutné opakovat tak dlouho, než bude splněna doba trvání). Podle datového listu diody jsem nastavil logickou nulu na hodnotu 0b11000000, logickou jedničku na hodnotu 0b11111000 a reset kód na 0b00000000. Společně se správným nastavením frekvence SPI periférie tyto kódy vyhovují časovým průběhům pro logické úrovně. V mém případě byla optimální frekvence 6 MHz. Poté jsem si vytvořil tři barevné profily. První odpovídá vypnuté diodě, druhý trvale svítící a třetí blikající. Profil je tvořen dvourozměrným polem. Jednotlivé profily je možné upravit pomocí funkce, kterou jsem vytvořil. Stačí do parametru funkce zapsat hodnotu pro červenou, zelenou a modrou. Pro nastavení diod jsem vytvořil druhou funkci, která sestaví pole dat, tak aby bylo možné nastavit barvu jedné diody podle vybraného profilu. Zbylé diody jsou vypnuty. Konstruování pole probíhá tak, že se nejdříve vytvoří 160 bytů resetovacího kódu, což odpovídá délce jeho trvání, poté se nastaví všechny LED diody na první profil až na diodu, která je vložena jako parametr funkce, ta je nastavena na profil, který je definovaný v druhém parametru. Pro odlehčení mikrokontroleru a ušetření času, byla pro vkládání dat do odesílacího registru SPI použita periférie DMAC (periférie pro přímý přístup k paměti). Ta za MCU přesouvá data z paměti do periférie. Kdyby nebyla použita, musel by mikrokontroler přesouvat data z paměti do periférie sám a tím zbytečně plýtvat výpočetní výkon. Úlohu jsem nastavil na DMAC kanál 0 a jako spouštěč jsem nastavil SERCOM0_Transmit. Což znamená, že když jsou k dispozici data a SPI je schopna data odeslat, spustí se přenos. Povolil jsem přerušování, pomocí kterého je DMAC schopen informovat o chybách a dokončení přenosu. Jako akci jsem nastavil přenos na jeden beat (což je minimální přenositelná informace) na DMA požadavek. Dále jsem nakonfiguroval, že se zdrojová adresa po každém přenosu inkrementuje a cílová adresa vždy zůstává stejná. Velikost beatu je nastavena na 8 bitů. V kódu se nejdříve pomocí funkce „DMAC_ChannelCallbackRegister“ definuje funkci zpětného volání. Jako této funkce parametry jsou: kanál DMAC, funkci zpětného volání a kontext. Požadavek na přenos dat se vytvoří pomocí funkce „DMAC_ChannelTransfer.“ Parametry této funkce jsou DMAC kanál, zdrojová adresa, cílová adresa a velikost dat. Po dokončení přenosu dat nebo chybě proběhne

přerušení a zavolá se funkce zpětného volání. Zde je definováno, co se má, v jakém případě stát, například se nastaví příznaky.



Obrázek 62: Logická 0 a logická 1



Obrázek 61: Průběh komunikace – reset signál

11.7STC3115IQT

STC3115IQT je integrovaný obvod pro měření parametrů Lithium-ion a Lithium Polymer akumulátorů. Podporuje dvě metody měření stavu nabití baterie. Prvním je měření napětí. Toto měření je méně přesné, ale je dostupné i v pohotovostním režimu. Druhým je měření proudu. Toto měření je přesnější, ale více energeticky náročné. Lze měřit stav nabití baterie buď pouze cestou měření napětí anebo oběma současně. V čipu je zabudovaný teplotní senzor, který hlídá teplotu baterie a celého zařízení. Pro přesné měření stavu nabití baterie je potřeba napětí nastavit registr `REG_VM_CNF`. Hodnota zapsaná do registru se vypočítá z vnitřního odporu akumulátoru R_i a jeho kapacity C_{nom} . Je dána rovnicí (1):

$$REG_VM_CNF = R_i * C_{nom} / 977,78. \quad (1)$$

Pro nastavení parametrů baterie pro měření proudu se musí nastavit registr `REG_CC_CNF`. Ve vzorci je použita hodnota odporu bočnicku R_{sense} a nominální kapacity akumulátoru C_{nom} . Hodnota zapsaná do registru se vypočítá podle vzorce (2):

$$REG_CC_CNF = R_{sense} * C_{nom} / 49,556. \quad (2)$$

Hodnota nabití baterie je uložena v registru REG_SOC. Jedná se o dvoubytový registr s adresami 0x02 a 0x03. V registru s nižší hodnotou adresy je uložena hodnota s nižší hodnotou. Měřič podporuje alarm, při nízké hodnotě nabití baterie a nízkém napětí. Tyto hodnoty jsou nastavitelné v registrech. Pokud dojde k překročení jedné z těchto mezí, je pin uzemněn. Komunikace mezi měřičem a MCU probíhá přes sériovou sběrnici I²C. Obvod podporuje režimy až do fast mode. Bity jsou v jednotlivých registrech zapsány tak, že první je MSB. Integrovaný obvod má 7bitovou klientskou adresu 0x70.²⁷

²⁷ (STMicroelectronics, 2018)

12 NÁVRH SOFTWAREVÉHO ŘEŠENÍ

12.1 Blokové schéma softwarového řešení projektu



Obrázek 63: Blokové schéma softwarového řešení

Softwarové řešení se skládá ze 3 základních bloků, a to firmware zařízení, Miotiq platformy a uživatelské aplikace.

Základní firmware zařízení zajišťuje obsluhu senzorů a aktuátorů na desce plošných spojů. Je nahrán na 32bitovém MCU PIC32CM JH. To se stará o chod boxu (odesílání/příjem dat, získání dat o akumulátoru, sběr dat z kapacitních senzorů pomocí periferie PTC, signalizace). Původně bylo zamýšleno použít dva mikrokontrolery. Prvním a hlavním mikrokontrolerem měl být ATSAML21J18B a druhým 8bitový AVR32DA32. Pro dva kontroléry jsme se původně rozhodli z důvodu nedostatku pinů na ATSAML21J18B. Od původní myšlenky bylo odstoupeno z důvodu použití programovatelné LED, jejíž barva je konfigurovatelná pomocí protokolu WS2815. Z důvodu použití této sériové komunikace a možnosti zapojit diody sériově, tak se výrazně snížil počet pinů, které by byly potřebné pro spínání dvojbarevných LED. Změna hlavního MCU byla učiněna z důvodu nedostatku pinů, které jsou kompatibilní s periferií PTC.

Miotiq platforma slouží k soustředění zpráv poslaných přes síť NB-IoT a k přeposlání do uživatelské aplikace. K přenosu dat ze zařízení do Miotiq platformy se používá UDP protokol. Platforma data dále přeposílá do aplikace přes HTTPS protokol.

Aplikace je sestavena z relační databáze, webového serveru a PHP serveru. V databáze obsahuje tabulky se záznamy. Data z databáze jsou po přihlášení do aplikace zobrazena. Uživatel má v aplikaci možnost nastavit zařízení a uložit si informace o aktuálně užívaných lécích. Webový server propojený s PHP serverem obsahuje všechny soubory potřebné pro vizualizaci a správnou funkci aplikace. Hlavní funkce, která je na straně serveru prováděna je komunikace s databází a platformou Miotiq.

12.2 Firmware

12.2.1 Původní koncept

Původně měl být firmware nahrán na dvou MCU. Prvním byl 32bitový ATSAML21J18B, na kterém měl běžet hlavní firmware a druhým 8bitový AVR128DA32, který měl obsluhovat měření přítomnosti léku v políčku pomocí kapacitního senzoru. Tyto dva mikrokontrolery spolu měli komunikovat pomocí I²C sériové komunikace. Řídicím zařízením na sběrnici mělo být 32bitové MCU ATSAML21J18B a klientem 8bitové AVR128DA32.

Firmware v 32bitovém mikrokontroleru měl být stejný jako aktuální s tím rozdílem, že by neprováděl měření kapacity, ale pomocí I²C komunikace by naměřená data vyčítal ze sekundárního.

Firmware na sekundárním 8bitovém MCU se měl starat o měření kapacity na kapacitních senzorech pomocí periferie PTC a odesílání dat primárnímu, pokud o ně požádá. Firmware na toto MCU měl být psán jako bare metal. Když by MCU neměřil kapacitu nebo by neprobíhala I²C komunikace, měl být ve spánkovém pohotovostním režimu. Z tohoto režimu by byl probuzen po přerušení generovaných periférií I²C při startu komunikace a zkontrolování adresy klienta. Poté by se probudil a zahájil měření kapacitních senzorů pomocí PTC periferie. Do doby, než se měření dokončí, by byla komunikace pozastavena. Data z kapacitních senzorů by byla uložena do příslušných registrů a poté by I²C komunikace pokračovala. Po jejím skončení by byl mikrokontroler znovu uveden do spánkového režimu.

12.2.2 Finální návrh firmware

Nakonec jsme se rozhodli kvůli již popsaným důvodům implementovat celou funkčnost do jednoho mikrokontroleru. MCU obsluhuje externí periferie. Jednou z externích periférií je integrovaný obvod pro měření parametrů akumulátoru, který komunikuje přes I²C sběrnici, druhou je NB-IoT modul komunikující přes sériovou sběrnici UART a třetí jsou programovatelné RGB LED s komunikačním protokolem WS2815. Je v něj implementován operační systém FreeRTOS, který obsluhuje vlákna, které obsahují stavové automaty, které obsluhují zmíněné periferie a aplikační funkce. Každé vlákno má jinou důležitost v důsledku vykonávané činnosti.

Firmware zařízení je psán na operačním systému CMSIS-FreeRTOS. Při inicializaci se provede základní nastavení a nakonfigurování integrovaného obvodu pro měření parametrů baterie. Po dokončení inicializace se vytvoří plánovač a vytvoří 5 úloh. První úlohou je APP_Tasks. Tato úloha obsahuje kód pro provádění aplikačních procesů. Její priorita je 2, což znamená, že je druhá nejvyšší. Mezi tyto procesy patří světelná a zvuková signalizace vybrání léku a světelná indikace procentuálního stavu nabití akumulátoru. Stav nabití akumulátoru je zobrazován třemi barevnými LED. Zelená dioda indikuje, že je stav nabití mezi 61 % až 100 %, mezi 21 % a 60 % a červená mezi 0 % až 20 %. Jedna z diod se rozsvítí po stisknutí postranního tlačítka a

zůstane svítit po dobu 5 s. Stisknutí tlačítka vygeneruje externí přerušeni, které probudí mikrokontroler a spustí tuto úlohu. V ní se nastaví hodnota pinu, na které je zapojena jedna z LED a zapne se jednorázový odpočet časovače. Po dokončení dojde k zablokování úlohy. Při jeho vypršení dojde k vygenerování přerušeni, probuzení MCU a zapnutí úlohy. V ní se deaktivuje světlo emitující dioda a úloha se znovu zablokuje. Čas pro upozornění na užití léku je kontrolován pomocí periferie RTC. Ta je nastavena do režimu kalendáře, který při shodě aktuálního času s časem alarmu vygeneruje přerušeni. Po spuštění zařízení probíhá synchronizace času RTC s časem aplikace. Stejně tak se synchronizují i časy alarmů. V případě, že je nastaveno užívání jednoho léku denně, tak je jako aktivní nastavena horní řada políček. Pokud jsou alarmy v aplikaci přenastavena, jsou tato ihned po jejich obdrženi aktualizována. Upozornění na užití léku probíhá tak, že když nastane RTC alarm, tak se spustí zelené podsvíceni políčka. Ve funkci zpětného volání je napsán jednoduchý stavový automat, ve kterém se vyhodnotí počet léků denně a pomocí nějž dojde k nastavení příznaku odpovídajícímu aktuálnímu času a dni. Po jeho nastavení je v aplikační části spuštěn stavový automat volající funkci nastavující barvu diody a zvukové upozornění. Nejprve se RGB dioda rozsvítí zelenou barvou a spustí se časovač na 10 minut, pokud ani po 10 minutách není lék vyjmut z políčka, RGB LED změní barvu na červenou a začne blikat. Aby se upozornění umocnilo, tak je blikání doplněno o zvukovou signalizaci. Poté, co dojde k odebrání léku z políčka, upozornění ustane a časovače se zastaví.

Druhou úlohou je NBIOT_TASKS_Tasks. Ta má obsluhuje periferii UART v režimu kruhové fronty a zpracovává obdržené zprávy ve formě AT příkazů. Priorita této úlohy je 4. Jedná se o nejvyšší prioritu. Každý příkaz, který modul BC660 odešle končí znakem „\n“, který má ASCII hodnotu 10. Z toho důvodu probíhá výčet dat z kruhové fronty vždy do obdrženi tohoto znaku. Po jeho přečtení je tento textový řetězec zpracován. Jelikož se jedná o úlohu s nejvyšší prioritou, tak pokud nejsou ve frontě žádná nepřečtená data, je pozastavena, aby mohly probíhat zbylé úlohy.

Třetí úlohou je NBIOT_APP_Tasks. Tato úloha provádí inicializaci narrow band modulu BC660. Její priorita je 3. Tato úloha probíhá pouze při prvotním spuštění. Po jejím dokončení je úloha zablokována. Obsahuje stavový automat, který obsahuje prvotní resetování modulu, registrace APN sítě, deaktivování PSM, nastavení eDRX, otevření síťového socketu a odeslání požadavku o synchronizaci času, alarmů a dalších parametrů modulu. Stavový automat funguje tak, že po každém odeslaném příkazu čeká na notifikaci, která potvrzuje jeho přijmutí. Po obdrženi notifikace se přepne stav stavového automatu.

Čtvrtou úlohou je PTC_APP_Tasks. Její priorita je 3. Tato úloha provádí kapacitní měření pomocí periferie PTC. Toto měření je také doprovázeno I²C komunikací s integrovaným obvodem pro měření parametrů akumulátoru, ze kterého získává stav nabití akumulátoru. Kapacitní měření probíhá periodicky každou sekundu. Po skončení měření jsou naměřená data zpracována, tak aby se z nich určilo, jestli je v políčku lék, nebo ne. Jelikož je měření pomocí PTC problematické a není možné stoprocentně určit pro každý senzor hodnotu, při které je políčko prázdné. Tak je nejdříve nutné z aplikace získat poslední stav léků v políčkách a použít

je jako referenční hodnoty. Logický stav je určen změnou naměřených hodnot. Při kladné změně je nastaven stav 1, při záporné 0.

Poslední úlohou je SLEEP_Tasks. Tato úloha má nejnižší prioritu (1). Vykonává uvedení mikrokontroleru do spánkového pohotovostního režimu. Z tohoto režimu je probouzen přerušeními generovanými periferiemi. Jelikož má úloha nejnižší prioritu, tak je nutné všechny úlohy s vyšší prioritou zablokovat, aby se úloha vykonala. Poté co periferie některá z periferií vygeneruje přerušení, tak se aktivuje jedna úloha nebo více přidělených úloh.

Celá funkce je popsána vývojovými diagramy v příloze Příloha B.1: Firmware.

12.3 Řešení aplikace

12.3.1 Oracle cloud

Pro hostování webové aplikace byla použita služka Oracle Cloud. Jedná se o službu cloudové výpočetní techniky poskytovanou společností Oracle Corporation poskytující servery, úložiště, databáze a nastavení sítě. Podporuje placené i neplacené účty. Podporuje vytvoření vlastní instance, která běží buď jako virtuální počítač, nebo jako bare metal. Virtuální zařízení jsou k dispozici buď na náhodném sdíleném počítači, nebo na vlastním, kde běží pouze aplikace jednoho uživatele. K dispozici je několik obrazů operačních systémů (Ubuntu, Windows, Oracle Linux, ...), ale je také možnost nahrát vlastní. Dále Oracle Cloud podporuje vytvoření blokového úložiště s podporou zálohy a zaváděcího úložiště pro uložení operačního systému. Pro propojení několika instancí, úložišť nebo databází a nastavení síťových bran, směrovacích tabulek a bezpečnostního seznamu podporuje virtuální cloudové síť. Ty je možné nastavit buď jako veřejné, nebo privátní. Oracle Cloud dále poskytuje několik druhů databází jako je Oracle databáze, MySQL, PostgreSQL a NoSQL

12.3.2 Ubuntu a Docker

Celá aplikace je provozována na operačním systému Ubuntu. Jedná se o volně šiřitelnou Linuxovou distribuci založenou na Debianu. Je oficiálně vydávána v několika edicích, konkrétně Desktop, Server a Core (pro zařízení internetu věcí a roboty). Operační systém je vyvíjen britskou firmou Canonical a komunitou jiných vývojářů. Poslední vydanou verzí je 23.10 a poslední verzí s dlouhodobou podporou je 22.04. Stejně jako ostatní linuxové distribuce je možné Ubuntu provozovat na samotném počítači, nebo na virtuálním stroji. Nová verze je vydávána každých 6 měsíců a verze s dlouhodobou podporou každé 2 roky. Canonical vydává bezpečnostní aktualizace a podporu vydání do konce životnosti.²⁸

Pro lepší oddělení aplikací běžících na serveru je na Ubuntu nainstalován balíček Docker. Jedná se o otevřenou platformu pro vývoj aplikací. Poskytuje možnost zabalit a spustit aplikaci v izolovaném prostředí zvaném kontejner. Izolace a zabezpečení dovoluje spustit mnoho kontejnerů současně na jednom zařízení. Aplikační Kontejnery jsou jednoduché na implementaci a obsahují všechno potřebné pro běh aplikace, takže není nutné se zabývat, co je nainstalováno na zařízení. Použití Dockeru dává jistotu, že když kontejner funguje na jednom zařízení, tak bude fungovat i na jiném, protože není závislý na použitém hardwaru.

Docker používá klient-server architekturu. Klient komunikuje s démonem, který provádí budování, spouštění a distribuování kontejnerů. Klient a démon můžou běžet na stejném systému, nebo je možné se připojit klientem ke vzdálenému démonu. Komunikují spolu pomocí

²⁸ (Canonical Ltd., Nedatováno)

REST API, přes UNIX sockety nebo síťové rozhraní. Dalším klientem je Docker Compose, který pomáhá pracovat s aplikacemi skládajících se z více kontejnerů.

Démon naslouchá API požadavkům a spravuje objekty Dockeru jako obrazy, kontejnery, sítě a svazky. Může také komunikovat s ostatními démony pro spravování služeb Dockeru.

Klient je primární způsob, kterým mnoho uživatelů komunikují s Dockerem. Když uživatel používá příkazy jako „docker run,“ klient odešle tyto příkazy démonu, který je předá dál. Tyto příkazy používají API. Klient může komunikovat s více než jedním démonem.

Při používání Dockeru uživatel vytváří a používá obrazy, kontejnery, sítě, svazky, pluginy a ostatní objekty. Obraz je šablona s instrukcemi pro vytvoření kontejneru. Často je založen na jiném obrazu, ale s nějakou dodatečnou úpravou. Například obraz webového serveru Apache je založen na obrazu Ubuntu s dodatečnou konfigurací potřebnou pro provoz aplikace. Uživatel může vytvořit i vlastní obraz. K tomu potřebuje vytvořit vlastní Dockerfile s jednoduchou syntaxí pro definování kroků potřebných pro vytvoření a spuštění obrazu. Každá instrukce v Dockerfile vytvoří vrstvu v obrazu. Když dojde ke změně souboru a restartování obrazu, jsou aktualizovány pouze vrstvy, které byly opraveny. Kontejner je běžící instance obrazu, kterou lze vytvořit, spustit, vypnout, přesunout nebo smazat pomocí Docker API nebo CLI. Kontejner jde připojit k jedné, nebo více sítím, přiřadit mu úložiště, nebo vytvořit nový obraz založený na jeho aktuálním stavu. Ve výchozím stavu jsou kontejnery relativně dobře izolovány od ostatních a hostujícím zařízení. Uživatel je schopen nastavovat, jak moc jsou sítě, úložiště a jiné subsystemy kontejneru izolovány. Kontejner je definován vlastním obrazem stejně jako konfigurací, která mu je nastavena při vytvoření a zapnutí. Po odstranění jsou všechna data, která nejsou odolná proti odstranění smazána.²⁹

12.3.3 Portainer

Pro spravování kontejnerů a obrazů Dockeru používáme aplikaci Portainer. Jedná se o službu pro organizování kontejnerů pomocí grafického uživatelského rozhraní. To pomáhá uživateli pracovat s kontejnery jednodušeji. Instalace je jednoduchá. Po instalaci Dockeru se stáhne obraz, vytvoří a spustí se kontejner. Lze zde vytvářet a spravovat kontejnery a obrazy, vytvářet zásobníky s kontejnery a vytvářet svazky pro ukládání jejich dat. Při vytváření nového kontejneru se ve webovém rozhraní nastavují stejné parametry jako při vytváření pomocí CLI. Například přiřazení paměťového svazku, vytvoření, nebo přiřazení sítě mezi kontejnery, nastavení podmínek restartování a další.

12.3.4 MySQL

Pro ukládání dat je využitý MySQL, což je open-source relační databázový systém. Je ideální pro malé i velké aplikace. Relační databáze organizuje data do jedné nebo více tabulek, ve

²⁹ (Docker Inc., Nedatováno)

kterých mohou být data propojena. SQL je jazyk, který programátoři používají k vytvoření, upravování a získávání dat z relační databáze. V tabulkách jsou data uspořádána do sloupců a řádek. Každý sloupec má přiřazený název a datový typ. V řádcích jsou uloženy jednotlivé záznamy. Datové typy mohou být například číselné, textové a datum a čas. Ty se dále dělí na jednotlivé datové typy, které se od sebe liší hlavně velikostí. Každému sloupci mohou být přiděleny další vlastnosti. Vlastnost NULL definuje, jestli může být sloupec nevyplněn. Dalším parametrem je AI („Auto Increment“), ten definuje, jestli se mají hodnoty sloupce automaticky zvyšovat. Toto je nejčastěji použito u identifikačního čísla řádku. Parametrem každé tabulky by měl být primární klíč. Ten je jednoznačným identifikátorem záznamu v tabulce. Každý primární klíč nesmí obsahovat hodnotu NULL a musí být pro každou tabulku definován právě jednou. Hodnota primárního klíče musí být v tabulce jedinečná.^{30 31}

Databázový systém MySQL byl nainstalován na server jako kontejner Dockeru z veřejně přístupného obrazu. Pro správu je použit obraz Adminer. Ten umožňuje jednoduchou správu MySQL databáze pomocí webového rozhraní.

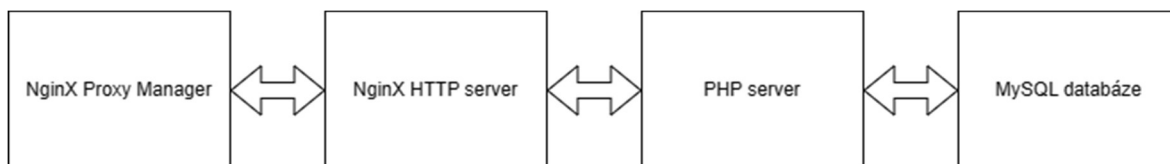
³⁰ (Oracle, Nedatováno)

³¹ (Wikipedie, 2022)

12.4 Webová aplikace

Webová aplikace se skládá z front-end (vizualizační části), a back-end (funkční části). Front-end obsahuje webových stránek, které jsou napsány v značkovacím jazyce HTML, CSS na vytvoření designu a JavaScriptu s knihovnou jQuery pro vytvoření dynamické stránky. Jako CSS šablona je použit Bootstrap. Back-end webové stránky se skládá z PHP souborů, jež zpracovávají data, ukládají do databáze, nebo vyzvedávají z databáze. Databáze je typu MySQL. Jsou v ní vytvořeny dvě tabulky, jež obsahují jednotlivé záznamy. Každý záznam v tabulce odpovídá jednomu zařízení nebo účtu.

Front-end webové aplikace se skládá z přihlašovací stránky, jež obsahuje formulář pro přihlášení. Na stejné stránce jsou uvedeny odkazy pro registraci, či resetování hesla. Na stránce pro registraci nového uživatele uživatel vyplní svou mailovou adresu a dvakrát heslo, aby předešel nesprávnému zadání. Pokud uživatel zapomene přihlašovací údaje, může si nechat na mailovou adresu poslat odkaz na změnu hesla. Po přihlášení je uživatel přesměrován na webovou stránku s vizualizací a ovládáním zařízení. Je zde vizualizace políček boxu, zobrazení stavu nabití akumulátoru, telefonní číslo na lékaře a informace o léku. Vizualizace stavu léků je tvořena pomocí jednoduché tabulky, jejíž políčka mění podbarvení podle stavu. Procentuální stav akumulátoru je zobrazen pomocí postupové lišty. Na pravé straně webové stránky jsou umístěny ovládací prvky jako formulář pro nastavení počtu léků za den, časů výběrů léků, telefonního čísla na lékaře a informace o léku. V horní části stránky je umístěna navigační lišta, která umožňuje přesměrování na zbylé stránky. Aplikace obsahuje mimo přihlašovacích stránek také domácí stránku, stránku nastavující identifikační číslo zařízení a stránku sloužící ke změně hesla.



Back-end webové aplikace se skládá ze 4 hlavních částí. První částí je NginX Proxy Manager. Pomocí této reverzní proxy klienti z internetu přistupují na HTTP server. Je tím zvýšeno zabezpečení webového serveru, který má přidělenou pouze interní IP adresu. Dalším benefitem jejího použití je možnost správy a obstarání SSL certifikátů. Nadále představuje možnost mít pod jednou veřejnou IP adresou a portem více http serverů. Druhou částí je NginX HTTP server, který slouží k odbavování HTTP požadavků. Tyto požadavky zpracovává a poté odešle odpověď obsahující obsah požadovaného souboru. Pro schopnost generování dynamického obsahu stránek aplikace obsahuje také PHP server. HTTP server je nastaven tak, aby požadavky směřující na soubory s příponou PHP na něj přesměroval. Jejich úložiště je propojeno. Ve sdílené paměti jsou uloženy soubory potřebné k zobrazení webové aplikace a pro komunikaci s databází a zařízením. Když PHP server obdrží požadavek od HTTP serveru, provede kód, který je obsažen v požadovaném souboru. Vygeneruje výsledný dynamický obsah, který může obsahovat HTML, JSON a jiné formáty. Tato data pošle zpět HTTP serveru, jenž je přepošle

klientovi. Poslední částí je MySQL databáze. Jedná se o relační databázi. V aplikaci databáze obsahuje dvě tabulky. První tabulka obsahuje uživatelské účty, konkrétně e-mailové adresy a hesla (hesla jsou zahashovaná). Druhá tabulka obsahuje samotná data z boxu. Tato data jsou po obdržení nových dat aktualizována.³²

V MySQL systému je jsem vytvořil databázi „spbox“ a v ní vytvořil dvě tabulky „accounts_tab“ a „spbox_tab.“ První zmiňovaná tabulka obsahuje informace o uživatelských účtech. Je složena ze 4 sloupců. První sloupec obsahuje identifikační číslo záznamu. Má vlastnost *primárního klíče* a *automatického zvyšování*. Je typu *integer*. Druhý sloupec obsahuje e-mailovou adresu a je typu *tinytext*. Třetí sloupec obsahuje hash uživatelského hesla, datový typ je *text*. V posledním sloupci je uložen token (neboli klíč k ověření identity) pro resetování zapomenutého hesla. Jedná se o náhodný textový řetězec. Který je ověřován, při odeslání požadavku na změnu hesla. Token je odeslán HTTP metodou GET. Tento sloupec má datový typ *text* a vlastnost *NULL*, takže může zůstat prázdný. Druhá tabulka obsahuje data o boxu na léky. Obsahuje celkem 32 sloupců. První sloupec obsahuje identifikační číslo boxu, má parametr *primárního klíče* a je datového typu *integer*. Může nabývat pouze *kladných* hodnot. Druhý sloupec obsahuje číslo, které určuje počet léků, které uživatel užívá denně. Je datového typu *tinyint* a může dosahovat pouze *kladných* hodnot. Následující tři sloupce obsahují čas. Tyto časy definují čas, ve který jsou léky vybírány. Jsou datového typu *time* (čas). Následuje 21 sloupců obsahujících stav políčka. Jsou datového typu *bit*. Když je lék v políčku jsou nastaveny na 1, jinak na 0. Poté tabulka obsahuje sloupce, ve kterých je uložen stav akumulátoru boxu (*tinyint*, pouze *kladné* hodnoty), telefonní číslo (*tinytext*), informace o léku (*text*), email (*tinytext*, *NULL*) uživatele pro propojení s uživatelským účtem a sloupce obsahující informace pro komunikaci se zařízením. První sloupec obsahuje IMSI (*tinytext*, *NULL*) a druhý URL (*text*, *NULL*), kam se mají data poslat. Více informací o složení tabulek je obsaženo v příloze Příloha C: Tabulka obsahu tabulek v databázi.

Aplikace je rozdělena na 3 části. První částí jsou datové soubory, druhou částí jsou vizualizační soubory a třetí assets (soubory získané z webového serveru pro zobrazení stránky), které obsahují multimediální soubory, css soubory a java skriptové skripty. Tyto assets mohou být prohlížečem uloženy do mezipaměti.

Datové soubory pracují s daty obsaženými v databázi. Zpracovávají data, která obdrží z HTTP požadavku a poté provedou interakci s databází. Jedním z těchto souborů je API pro komunikaci s koncovým zařízením. Data obdržená od koncového zařízení jsou v JSON formátu. Obsahují IMSI, IP adresu zařízení, porty a data, která je nutné dekodovat z formátu Base64. Pro snížení přenášeného objemu dat, jsou data přenášena v hexadecimálním formátu. Z toho důvodu je nutné převést každý znak na jeho decimální ASCII hodnotu. Jelikož box odesílá dva typy dat, tak je poté kontrolována délka obdržených dat, čímž je rozhodnuto, o jaký požadavek se jedná. Prvním požadavkem je žádost o synchronizaci boxu s daty v databázi a

³² (WEDOAPPS LLC., 2023)

aktuálním časem. Po obdržení tohoto požadavku, jsou zařízení zpět odeslána data, která jsou spojena s jeho identifikačním číslem. Druhým typem dat jsou naměřená data. Jednou hodnotou je ID zařízení, další stav nabití akumulátoru a zbylé tři hodnoty udávají stav léků v políčkách. Stav léků je kvůli snížení objemu odesílaných dat zakódován do tří bytových čísel, kde bit 0-7 symbolizuje hodnotu stavu. Hodnoty na 7. bitu obsahují hodnoty, které odpovídají políčku umístěnému úplně vlevo v každém řádku. Tato čísla jsou tedy pomocí bitových posunů dekodována a následně všechna uložena pod odpovídající identifikační číslo.

Zbylé API jsou uzpůsobené pro zpracování dat odeslaných z formulářů. Ty pracující s uživatelskými hesly tyto údaje získávají pomocí metody POST. Tato data jsou přenášena pomocí HTTPS, to znamená, že jsou zašifrovaná. Hesla jsou před uložením do databáze zahashována PHP funkcí `password_hash`. Ještě před vytvořením hashe je k zadanému heslu přidána náhodně vygenerovaná sůl. K porovnávání hesla s uloženým hashem je použita PHP funkce `password_verify`. V API k odeslání odkazu k resetování hesla uživateli na email je použita třída `PHPMailer`. Odkaz pro resetování hesla je vytvořen jako odkaz obsahující GET parametr `token`, jehož hodnota je náhodně vygenerovaný textový řetězec. Tento řetězec je uložen do tabulky k záznamu uživatele, aby mohl být po odeslání požadavku s tímto parametrem ověřen. API pracující s daty nastavující parametry boxu jako jsou počet léků denně a časy na vybrání léků odešle po jejich zpracování tyto data do boxu (jsou uložena také do databáze), kde jsou dále zpracována. Zbylé údaje jsou po jejich odeslání uloženy do databáze.

Celá funkce je popsána vývojovými diagramy v příloze Příloha B.2: Webová aplikace.

Vizualizační soubory jsou takové soubory, které jsou odeslány prohlížeči, který z nich sestaví zobrazení webové stránky. Jedněmi z nich jsou HTML soubory, v kterých jsou napsány značky definující hrubý obsah webové stránky. CSS soubory dodávají webovým stránkám vzhled a skripty `JavaScriptu` vytvářejí dynamický obsah na straně uživatele. Pro zjednodušení byla do projektu implementována knihovna `jQuery`. Pomocí jejích funkcí probíhá neustálé obnovování dat z boxu zobrazených na domácí stránce. Díky jejímu použití není nutné znovu načítat celou webovou stránku, ale pouze její části. Některé soubory obsahují také PHP kód, u nichž PHP server na základě obdržených parametrů vygeneruje HTML.

13 ZÁVĚR

Jako cíl projektu jsme si vytyčili, vytvořit chytrý organizér na léky s funkcí snímání přítomnosti medikamentů pomocí technologie kapacitních senzorů. I když od začátku nebyla schopnost detekování zcela jistá, nakonec se nám podařilo docílit z velké části funkčního produktu s poměrně spolehlivou detekcí. K tomu nám dopomohli také testovací desky, na kterých bylo možné otestovat co nejspolehlivější konfiguraci a design senzoru. Také implementace komunikační technologie NB-IoT se dá hodnotit úspěšně. Kvůli použitému poskytovateli Vodafone a nepodpoře ukládání nepřijatých dat na síťový server, není možné použít spánkové režimy, pro ještě větší úsporu energie bylo nutné jej změnit. Použitím funkce dočasněho uložení dat na síťový server by odpadla nutnost neustálého příjmu. Data by byla přijata po zažádání modulu. Synchronizace času by nebyla poznamenána, jelikož je po odeslání dat vždy spuštěno okno pro příjem zpráv. Díky jejímu velkému dosahu, je možné zařízení ovládat na dálku pomocí webové aplikace. Případným vylepšením produktu by mohla být hlavně samotná webová aplikace. V té by mohla být v budoucnu přidána funkce více boxů k jednomu účtu. Další přidanou funkcí by mohlo být upozornění přes některou z konverzačních sociálních sítí. Tento způsob komunikace se zařízením by mohl být privilegován lidmi, kteří chtějí nastavovat zařízení svým blízkým, nebo organizacemi, jež vlastní více zařízení a každé z nich mohou nastavit z jednoho místa. Pro uživatele, kteří si chtějí dávkování léků spravovat sami, by využití této komunikační technologie mohlo být zbytečné. Za zvážení by možná stálo vytvoření druhé verze, která by obsahovala bezdrátovou komunikaci s nízkým dosahem. Jednou z variant by mohl být Bluetooth. S tím by se naskytla nutnost vytvořit mobilní aplikaci, přes kterou by bylo zařízení nastavováno a sledováno.

I když je detekování kapacitními senzory vcelku spolehlivé, tak se nám léky zatím podařilo detekovat pouze pokud krabička nebyla více nakloněna, nebo otočena vzhůru nohama. Pro vytvoření komplexnějšího snímání by bylo nutné vytvořit takový design senzoru, který bude tyto mínusy schopen co nejvíce eliminovat. Tento problém by mohl být potlačen také pozměněním designu instalačního boxu. Pro snížení dopadů vnějších vlivů na výsledek snímání by bylo optimální využít v dalších verzích aktivní stínění. Z nabytých zkušeností z projektu bychom si dovolili říci, že kapacitní detekování může být cesta. Ale pro reálné nasazení do produktů by bylo potřeba ještě spousta práce.

Během prozatímního vývoje zatím nebylo dosaženo úplného zprovoznění integrovaného obvodu pro sledování akumulátoru. V aktuálním stavu je schopen měřit napětí mezi elektrodami článku. I přes několikanásobnou snahu zprovoznit měření parametru SoC (stav nabití akumulátoru v procentech) jsem zatím nebyl schopen přijít na důvod neaktivity.

14 SEZNAM TERMÍNŮ

NB-IoT – Narrowband Internet of Things, bezdrátová komunikační technologie

MCU – mikrokontroler

LED – světlo emitující dioda

LoRa – bezdrátová komunikační síť pracující v ISM pásmu s velkým dosahem, vyvinuta společností Semtech

LoRa WAN – komunikační protokol pro technologii LoRa

Bluetooth – bezdrátová komunikační technologie pracující v ISM pásmu s lokálním dosahem

RAM – proměnná paměť s náhodným přístupem

Flash – neproměnná paměť

SRAM – statická proměnná paměť s náhodným přístupem

DMA – přímý přenos dat

DMAC – ovladač pro přímý přenos dat

TC – časovač a čítač

TCC – časovač, čítač pro řídicí aplikace

UART, USART, SPI, I²C – sériová komunikace

ADC – analogově digitální převodník

PTC – periferie pro kapacitní měření

SWD – programovací rozhraní

Heap – halda

Boolean – datový typ nabývající hodnot pravda a lež

LTE Cat NB2 – verze NB-IoT

eSIM – virtuální SIM karta

UDP, TCP – transportní protokoly

MQTT, CoAP, http, HTTPS – aplikační protokol

DFOTA – aplikační protokol pro vzdálené nahrání firmware

SMS – služba krátkých textových zpráv

USB – univerzální sériová sběrnice

COM port – komunikační port

APN – brána mobilní sítě

URC – nevyžádaná zpráva

PDP – datová relace

Socket – koncový bod připojený přes síť

URL – jednotný lokátor zdroje

RGB – barevný model

RTC – hodiny reálného času

Back-end – datová vrstva

Front-end – prezenční vrstva

Démon – program obsluhující události

Hash – výstup kryptografické funkce

Token – elektronický klíč

IMSI – unikátní číslo přidělené SIM kartě operátorem

Asset – zdroj, který vede ke zhodnocení

ASCII – kódová tabulka obsahující znaky

Downlink – komunikace směřovaná ze serveru na zařízení

Uplink – komunikace směřovaná ze zařízení na server

15 SEZNAM OBRÁZKŮ

Obrázek 1: 3D vizualizace kompletního boxu.....	10
Obrázek 2: 3D vizualizace desky plošných spojů	11
Obrázek 3 - Blokové schéma zařízení	13
Obrázek 4: Blokové schéma projektu.....	13
Obrázek 5: Logo aplikace Fusion 360, dostupné z: https://in.pinterest.com/pin/268456827780030537/	14
Obrázek 6: Schéma zapojení headeru pro externí debugger a reset tlačítka	16
Obrázek 7: Schéma zapojení I2C filtru.....	17
Obrázek 8: Schéma zapojení externího krystalu.....	17
Obrázek 9: Mikrokontroler PIC32CM5164JH01064 v 64 pinovém pouzdře TQFP, dostupné z: https://www.microchip.com/en-us/product/pic32cm5164jh01064	18
Obrázek 10: Schéma zapojení MCU	19
Obrázek 11: NB - IOT modul, dostupné z https://www.soselectronic.com/cz/articles/quectel/vicepasmovy-lte-cat-nb2-modul-bc660k-gl-2593	20
Obrázek 12: Schéma NB - IOT modulu	21
Obrázek 13: Schéma zapojení magnetického bzučáku.....	22
Obrázek 14: Magnetický bzučák, dostupné z https://cz.mouser.com/ProductDetail/CUI-Devices/CMI-9653S-SMT-TR	22
Obrázek 15: Prototypovací deska pro kapacitní senzory	24
Obrázek 16: Druhá strana prototypovací desky pro kapacitní senzory	24
Obrázek 17: Jednotlivé vrstvy prototypovací desky pro kapacitní senzory	25
Obrázek 18: Finální deska plošných spojů, konkrétně 4 vrstva, stínění kapacitních senzorů ..	26
Obrázek 19: Baterie Li - Pol, dostupné z https://www.ampul.eu/cs/baterie/4694-li-pol-baterie-4000mah	27
Obrázek 20: Schéma převodníku napětí k programovatelným LED diodám.....	29
Obrázek 21: Podrobně rozkreslené schéma funkce převodníku napětí	29
Obrázek 22: Ukázka zapojení 12 z 42 programovatelných RGB LEDek	30
Obrázek 23: RGB programovatelné LED, dostupné z https://www.digikey.com/en/products/detail/inolux/IN-PI42TASPRPGPB/9681236	30
Obrázek 24: Schéma zapojení CHARGERU.....	32
Obrázek 25: Tabulka možných stavů chargeru, dostupné z https://cz.mouser.com/datasheet/2/268/MCHPS02791_1-2520625.pdf	32
Obrázek 26: Schéma zapojení Fuel Gauge	33
Obrázek 27: Stabilizátor TPS63021DSJT, dostupné z: https://www.ti.com/product/TPS63021/part-details/TPS63021DSJT	34
Obrázek 28: Schéma zapojení stabilizátoru na 3,3 V	35
Obrázek 29: Schéma zapojení neinvertujícího buck-boost měniče	35
Obrázek 30: Schéma stabilizátoru na 5V	36

Obrázek 31: Stabilizátor TPS61240DRVR, dostupné z: https://www.ti.com/product/TPS61240/part-details/TPS61240DRVR	36
Obrázek 32: USB C, dostupné z https://cz.mouser.com/ProductDetail/GCT/USB4105-GF-A-060?qs=QNEbhJQKvY2mocwNJzgfA%3D%3D	37
Obrázek 33: Schéma zapojení USB C konektoru s pojistkou	37
Obrázek 34: : Schéma zapojení některých signalizačních LED	38
Obrázek 35: Signalizační LED diody	39
Obrázek 36: Signalizační LED diody ovládané z mikrokontroleru.....	39
Obrázek 37: Signalizační LED dioda ovládaná z NB - IOT modulu	39
Obrázek 38: 1. vrstva finální desky	40
Obrázek 39: 2. vrstva finální desky	41
Obrázek 40: 3. vrstva finální desky	42
Obrázek 41: 4. vrstva finální desky	43
Obrázek 42: Jednotlivé vrstvy prototypovací desky pro kapacitní senzory s červeně vyznačenými částmi, které byli po výrobě zjištěni, že chybí	44
Obrázek 43: Screen z logického analyzátoru při analyzování sběrnice UART.....	45
Obrázek 44: Vizualizace modelované krabičky pohled shora.....	47
Obrázek 45: : Vizualizace modelované krabičky s šuplíčky	47
Obrázek 46: : Vizualizace modelované krabičky shora se šuplíčky.....	48
Obrázek 47: : Vizualizace modelované krabičky ze strany	48
Obrázek 48: Vizualizace modelované krabičky se zátkou na baterku a s PCB.....	49
Obrázek 49: Vizualizace modelované krabičky s PCB	49
Obrázek 50: : Vizualizace modelované krabičky se zátkou na baterku a s PCB.....	49
Obrázek 51: Logo první generace verze 1	50
Obrázek 52: Logo první generace verze 2	50
Obrázek 53: Logo druhé generace verze 1	51
Obrázek 54: Logo druhé generace verze 2	51
Obrázek 55: Ikona projektu k logu	51
Obrázek 56: Testování kapacitního senzoru pomocí zásuvného modulu data visualizer.....	56
Obrázek 57: Komunikace s modulem BC660	60
Obrázek 58: Node-RED výstup ladicího nodu	60
Obrázek 59: Logická analýza – komunikace s NB-IoT modulem BC660	61
Obrázek 60: Sekvence čipu WS2815. Získáno z datového listu diody IN-PI42TAS(X)R(X)G(X)B. Inolux Corp. Str. 5. 2021. Dostupné z: https://www.inolux-corp.com/datasheet/SMDLED/Addressable%20LED/IN-PI42TAS(X)R(X)G(X)B_v1.1.pdf	62
Obrázek 61: Průběh komunikace – reset signál.....	64
Obrázek 62: Logická 0 a logická 1	64
Obrázek 63: Blokové schéma softwarového řešení	66
Obrázek 64: Blokové schéma aplikace	73

16 CITOVANÁ LITERATURA

Amazon Web Services Inc. Nedatováno. Memory Management. *freertos.org*. [Online] Nedatováno. [Citace: 27. Říjen 2023.] <https://freertos.org/a00111.html>.

Microchip Technology Inc. 2009. [Online] 2009. [Citace: 3. Březen 2024.] https://cz.mouser.com/datasheet/2/268/MCHPS02791_1-2520625.pdf.

Arm Ltd. 2020. Arm Cortex-M0+ Processor Datasheet. [Online] 2020. [Citace: 27. Říjen 2023.]

Autodesk, Inc. Nedatováno. Autodesk Fusion 360. *Autodesk web*. [Online] Autodesk, Nedatováno. [Citace: 26. Únor 2024.] <https://www.autodesk.cz/products/fusion-360/overview?term=1-YEAR&tab=subscription>.

Canonical Ltd. Nedatováno. The story of Ubuntu. *ubuntu.com*. [Online] Nedatováno. [Citace: 9. Leden 2024.] <https://ubuntu.com/about>.

CUI Devices. 2019. CMI-9653S-SMT-TR Datasheet - Audio Indicators | Buzzers | CUI Devices. *CMI-9653S-SMT-TR Datasheet - Audio Indicators | Buzzers | CUI Devices*. [Online] 11. Listopad 2019. [Citace: 16. Březen 2024.] https://cz.mouser.com/datasheet/2/670/cmi_9653s_smt_tr-1777433.pdf.

Deutsche Telekom Inc. 2023. NB-IoT Network Information. *docs.iotcreators.com*. [Online] Srpen 2023. [Citace: 13. Prosinec 2023.] <https://docs.iotcreators.com/docs/nb-iot-network-information>.

Docker Inc. Nedatováno. Docker overview. *docs.docker.com*. [Online] Nedatováno. [Citace: 13. Leden 2024.] <https://docs.docker.com/get-started/overview/>.

Inolux Corp. 2021. IN-PI42TAS(X)R(X)G(X)B. *inolux-corp.com*. [Online] 2021. [Citace: 2023. Prosinec 12.] [https://inolux-corp.com/datasheet/SMDLED/Addressable%20LED/IN-PI42TAS\(X\)R\(X\)G\(X\)B_v1.4.pdf](https://inolux-corp.com/datasheet/SMDLED/Addressable%20LED/IN-PI42TAS(X)R(X)G(X)B_v1.4.pdf).

Inolux Corporation. 2018. IN-PI42TAS(X)R(X)G(X)B. *IN-PI42TAS(X)R(X)G(X)B*. [Online] 1. Červen 2018. [Citace: 17. Listopad 2023.] https://cz.mouser.com/datasheet/2/180/IN_PI42TAS_X_R_X_G_X_B_v1_1-3387395.pdf.

Kutý, Michael. Nedatováno. 2. Architektura počítače (von Neumannovo a Harwardské schéma, Flynnova taxonomie, základní deska, procesor, mikroarchitektura procesoru, paměti, sběrnice, řadič, přídatné karty, ovladače). *michaelkuty.github.io*. [Online] Nedatováno. [Citace: 27. Říjen 2023.] <https://michaelkuty.github.io/ssz-ai-hk-3/tech/2.html>.

Microchip Technology Inc. 2020. QTouch® Modular Library Peripheral Touch Controller. *Microchip*. [Online] 2020. [Citace: 5. Leden 2024.]

<https://ww1.microchip.com/downloads/en/DeviceDoc/QTtouch-Modular-Library-PTC-UG-DS40001986E.pdf>.

—. **2014.** AT02259: QTtouch Schematic and Layout Checklist. *AT02259: QTtouch Schematic and Layout Checklist*. [Online] Červenec 2014. [Citace: 15. Listopad 2023.] https://ww1.microchip.com/downloads/en/appnotes/atmel-42094-qtouch-schematic-and-layout-checklist_applicationnote_at02259.pdf.

—. **2023.** Introduction to QTtouch® Peripheral Touch Controller (PTC). *Developer Help*. [Online] Microchip Technology Inc., 9. Listopadu 2023. [Citace: 5. Únor 2024.] <https://developerhelp.microchip.com/xwiki/bin/view/applications/touch-gesture/atmel-studio-qtouch-library-composer-legacy/Intro-to-qtouch-ptc/>.

—. **2022.** PIC32CM JH00/JH01. *microchip.com*. [Online] 2022. [Citace: 27. Říjen 2023.] <https://ww1.microchip.com/downloads/aemDocuments/documents/MCU32/ProductDocuments/DataSheets/PIC32CM-JH00-JH01-Family-Data-Sheet-60001632.pdf>.

—. **2022.** PIC32CM JH00/JH01 Family Data Sheet. *PIC32CM JH00/JH01 Family Data Sheet*. [Online] 2022. [Citace: 10. 12. 2023.] <https://ww1.microchip.com/downloads/aemDocuments/documents/MCU32/ProductDocuments/DataSheets/PIC32CM-JH00-JH01-Family-Data-Sheet-60001632.pdf>.

—. **Nedatováno.** PIC32MZ0512EFE064. *Microchip web*. [Online] Microchip, Nedatováno. [Citace: 18. Březen 2024.] <https://www.microchip.com/en-us/product/PIC32MZ0512EFE064>.

—. **2020.** QTtouch® Modular Library Peripheral Touch Controller. [Online] 2020. [Citace: 19. Březen 2024.] <https://ww1.microchip.com/downloads/en/DeviceDoc/QTtouch-Modular-Library-PTC-UG-DS40001986E.pdf>.

Oracle. Nedatováno. What is MySQL? *dev.mysql.com*. [Online] Nedatováno. [Citace: 23. Leden 2024.] <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>.

Quectel Wireless Solutions Co. 2023. IoT Modules and Antenna Catalogue. *quectel.com*. [Online] 2023. [Citace: 3. Listopad 2023.] https://www.quectel.com/wp-content/uploads/2023/03/Quectel_Product_Brochure_V7.4.pdf.

ShawnHymel. Nedatováno. Getting Started with STM32 - Introduction to FreeRTOS. *digikey.cz*. [Online] Nedatováno. [Citace: 27. Říjen 2023.] <https://www.digikey.cz/en/maker/projects/getting-started-with-stm32-introduction-to-freertos/ad275395687e4d85935351e16ec575b1>.

STMicroelectronics. 2018. STC3115. *cz.mouser.com*. [Online] 2018. [Citace: 12. Prosinec 2023.] <https://cz.mouser.com/datasheet/2/389/stc3115-1850233.pdf>.

Texas Instruments Inc. 2021. FDC1004: Basics of Capacitive Sensing and Applications. [Online] 2021. [Citace: 18. Březen 2024.] https://www.ti.com/lit/an/snoa927a/snoa927a.pdf?ts=1710918578479&ref_url=android-app%253A%252F%252Fcom.google.android.googlequicksearchbox%252F.

Texas Instruments Incorporated. 2019. [Online] 2019. https://www.ti.com/lit/ds/symlink/tps63021.pdf?ts=1711298578101&ref_url=https%253A%252F%252Fhu.mouser.com%252F.

—. **2015.** [Online] 2015.

—. **2015.** I2C Bus Pullup Resistor Calculation. *Texas Instruments*. [Online] Únor 2015. [Citace: 10. Březen 2024.] https://www.ti.com/lit/an/slva689/slva689.pdf?ts=1711268799329&ref_url=https%253A%252F%252Fwww.google.com%252F.

Výukový portál COPTel. 2019. Kapacitní snímače. *Výukový portál COPTel*. [Online] 24. Listopad 2019. [Citace: 29. Listopad 2023.] <https://coptel.cz/mod/page/view.php?id=6692>.

WEDOAPPS LLC. 2023. How Does PHP Work With The Web Server And Browser? *wedowebapps.com*. [Online] 2023. [Citace: 24. Únor 2024.] <https://www.wedowebapps.com/php-work-with-the-web-server-and-browser/>.

Wikipedie, otevřená encyklopedie. 2024. Akumulátor. *Wikipedie, otevřená encyklopedie*. [Online] 1. 12 2024. [Citace: 3. Únor 2024.] <https://cs.wikipedia.org/wiki/Akumul%C3%A1tor>.

Wikipedia otevřená encyklopedie. 2022. Jednočipový počítač. *Wikipedia*. [Online] Wikipedia, 25. Říjen 2022. [Citace: 24. Březen 2024.] https://cs.wikipedia.org/wiki/Jedno%C4%8Dipov%C3%BD_po%C4%8D%C3%ADta%C4%8D.

Wikipedie, Příspěvatelé. 2022. Primární klíč. *cs.wikipedia.org*. [Online] 2022. [Citace: 23. Leden 2024.] https://cs.wikipedia.org/wiki/Prim%C3%A1rn%C3%AD_kl%C3%AD%C4%8D.

17 SEZNAM PŘÍLOH

Příloha A: Elektrická schémata zapojení

Příloha A.1: Uživatelské rozhraní

Příloha A.2: Napájení

Příloha A.3: MCU

Příloha A.4: Komunikační modul

Příloha A.5: RGB LED

Příloha A.6: Kapacitní senzory

Příloha B: Vývojové diagramy

Příloha B.1: Firmware

Příloha B.1.a: Obecný vývojový diagram

Příloha B.1.b: Úloha PTC_APP_Tasks

Příloha B.1.c: Úloha NBIOT_APP_Tasks

Příloha B.1.d: Úloha NBIOT_TASKS_Tasks

Příloha B.1.e: Úloha APP_Tasks

Příloha B.1.f: Úloha SLEEP_Tasks

Příloha B.2: Webová aplikace

Příloha B.2.a: rcv_data

Příloha B.2.b: verify

Příloha B.2.c: token

Příloha B.2.d: spass

Příloha B.2.e: cont

Příloha B.2.f: mreg

Příloha B.2.g: chpass1

Příloha B.2.h: setup1

Příloha B.2.i: setup2

Příloha B.2.j: setup3

Příloha C: Seznam obsahu tabulek v databázi

Příloha D: Výkresy krabice

Příloha D.1: Spodní díl krabice

Příloha D.2: Horní díl krabice

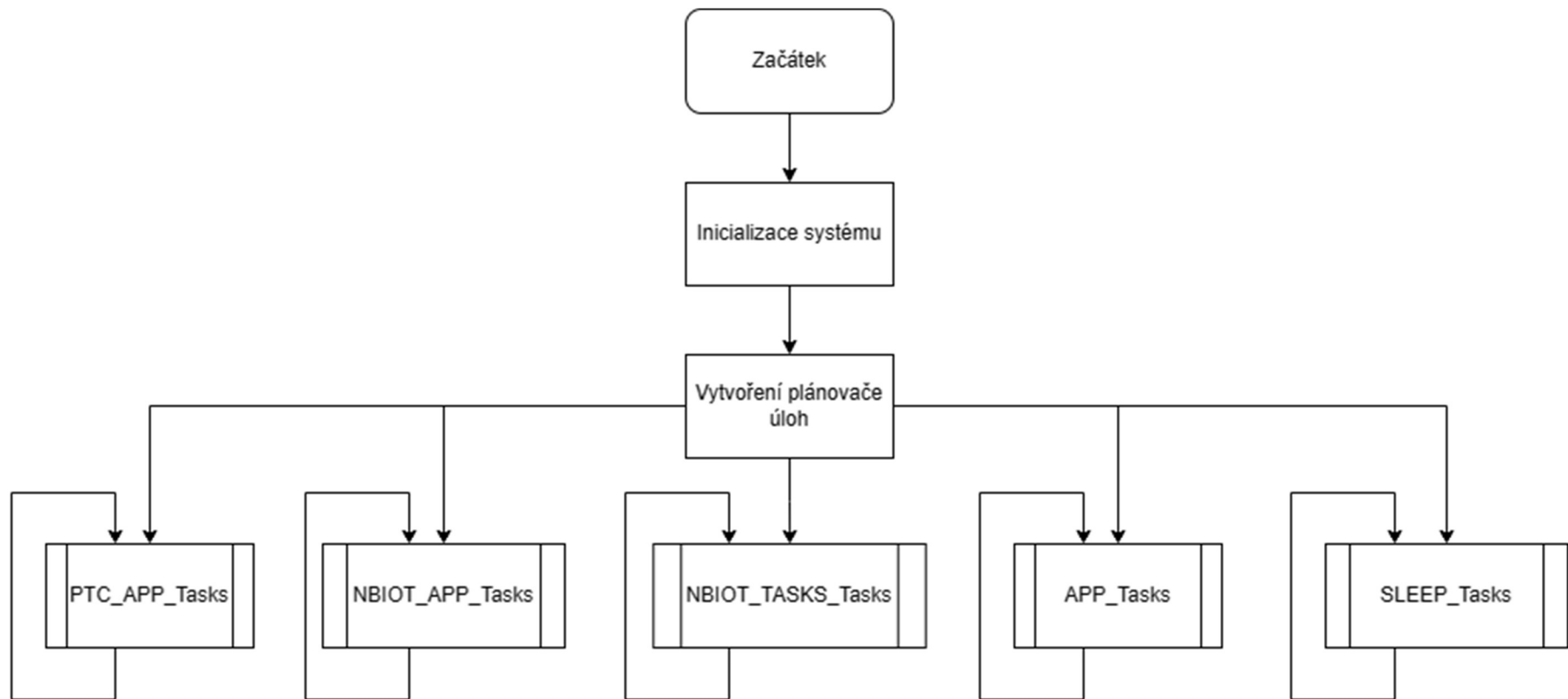
Příloha D.3: Krytka na prášky krabice

Příloha E: BOM list projektu

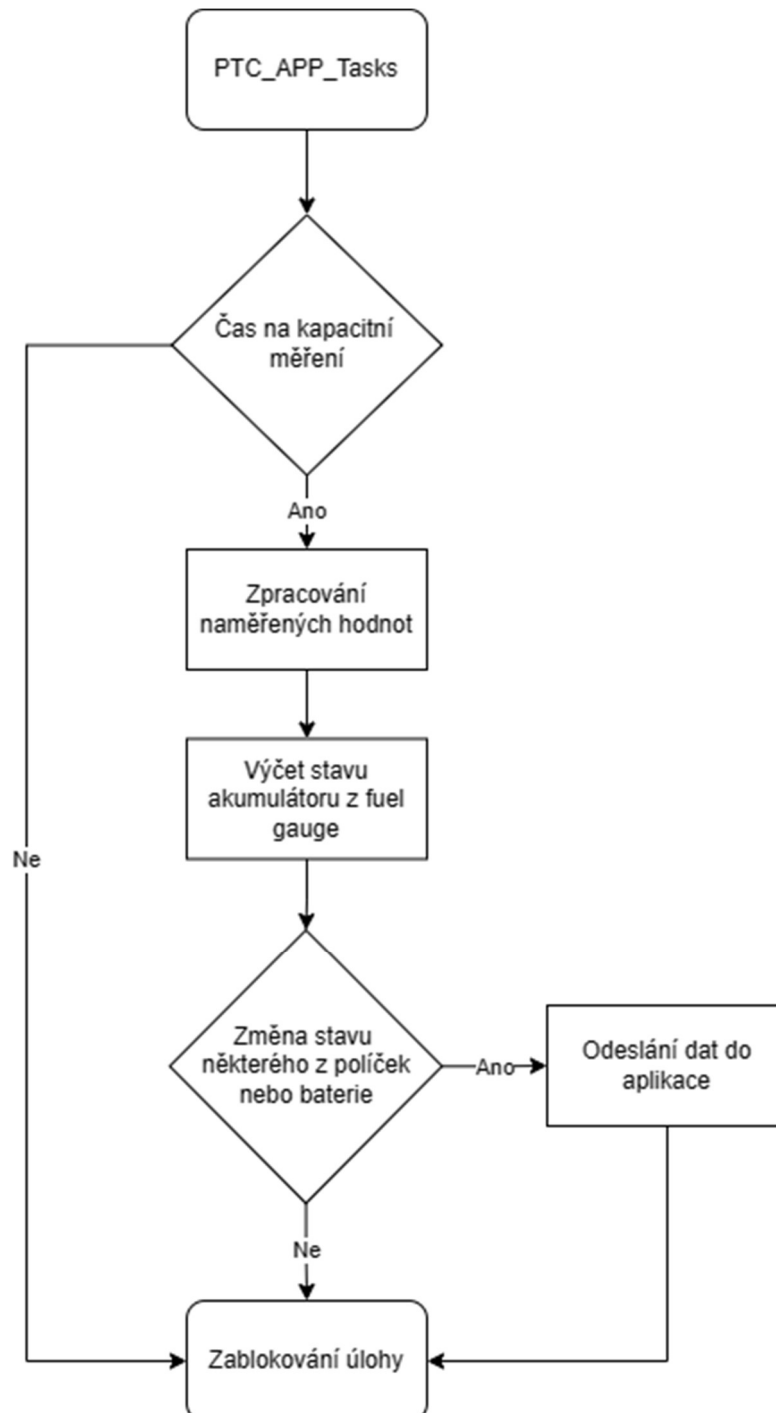
Příloha E.1: BOM list stránka 1

Příloha E.2: BOM list stránka 2

Příloha B.1.a: Obecný vývojový diagram



Příloha B.1.b: Úloha PTC_APP_Tasks



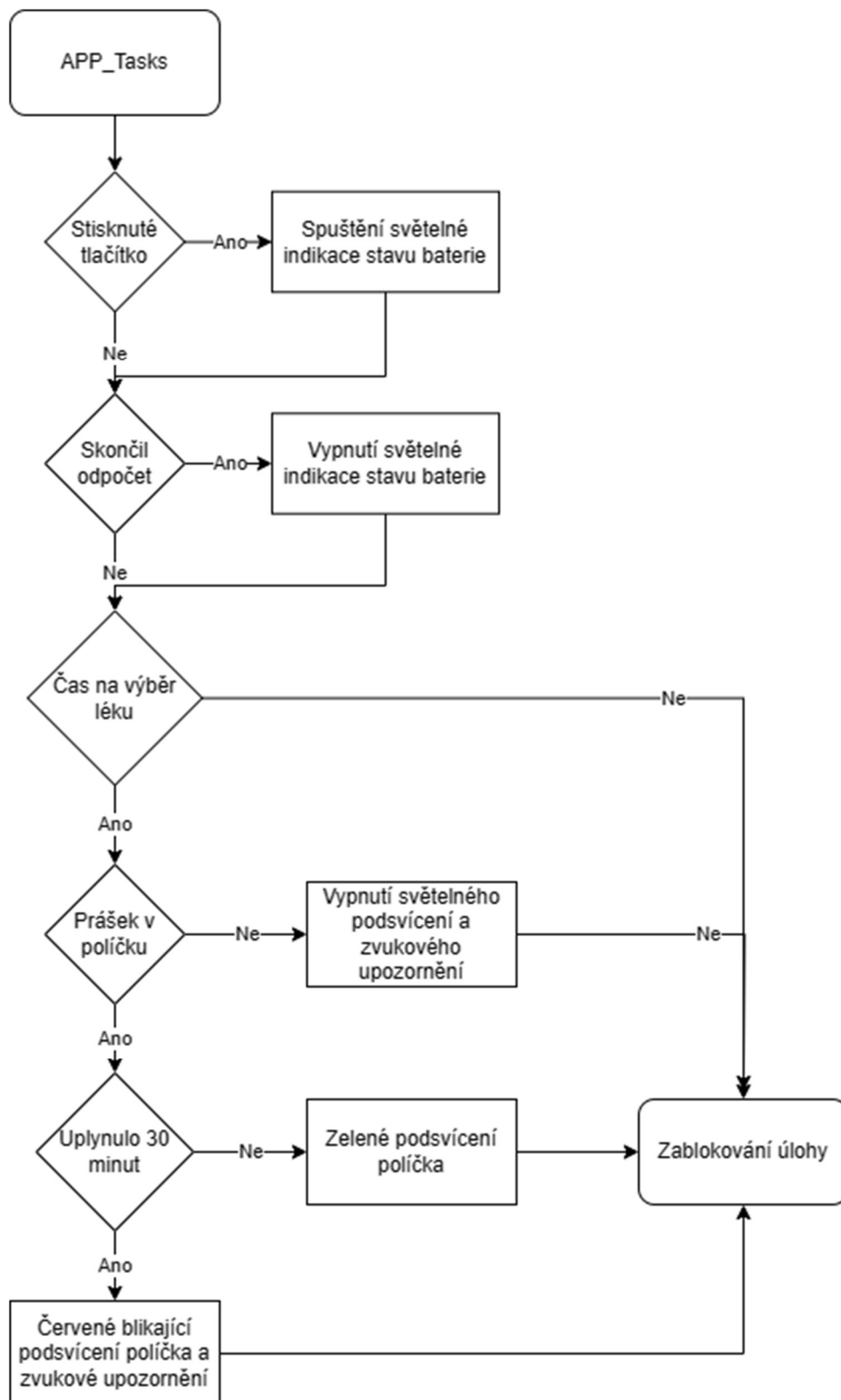
Příloha B.1.c: Úloha NBIOT_APP_Tasks



Příloha B.1.d: Úloha NBIOT_TASKS_Tasks



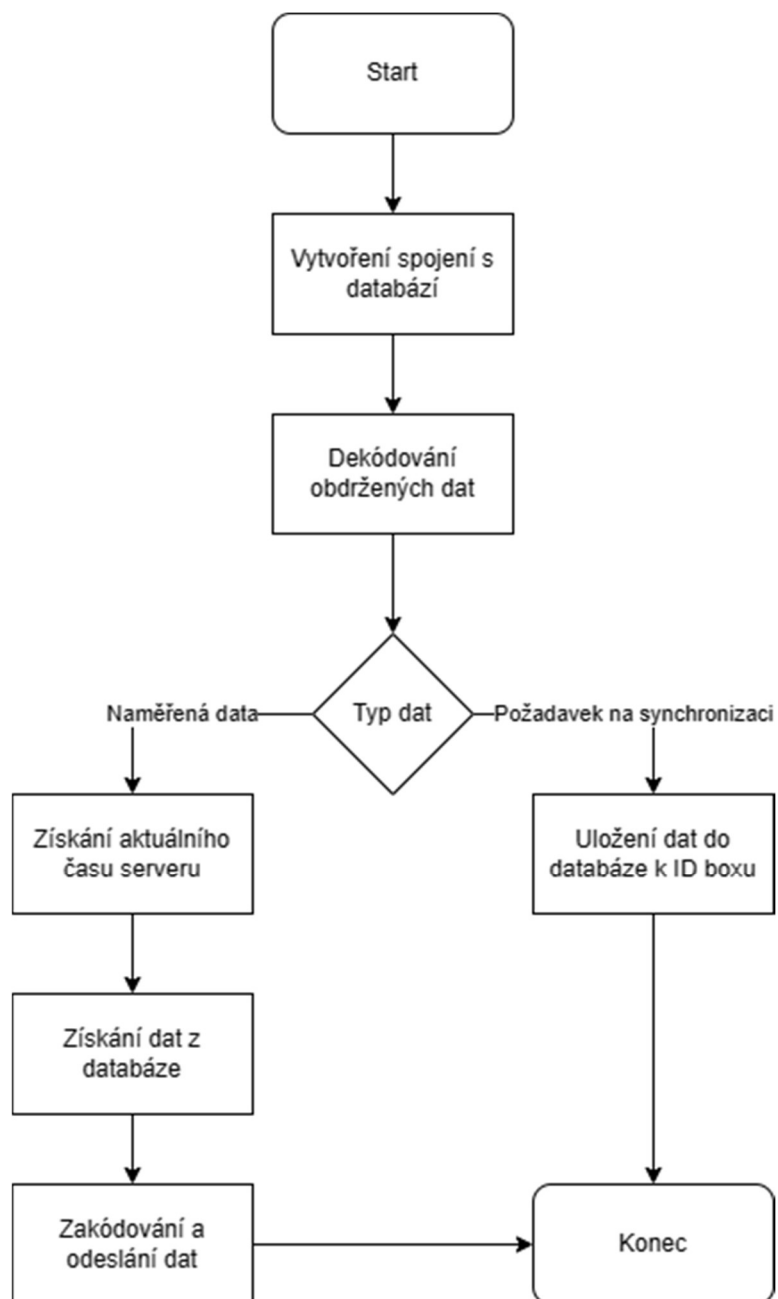
Příloha B.1.e: Úloha APP_Tasks



Příloha B.1.f: Úloha SLEEP_Tasks



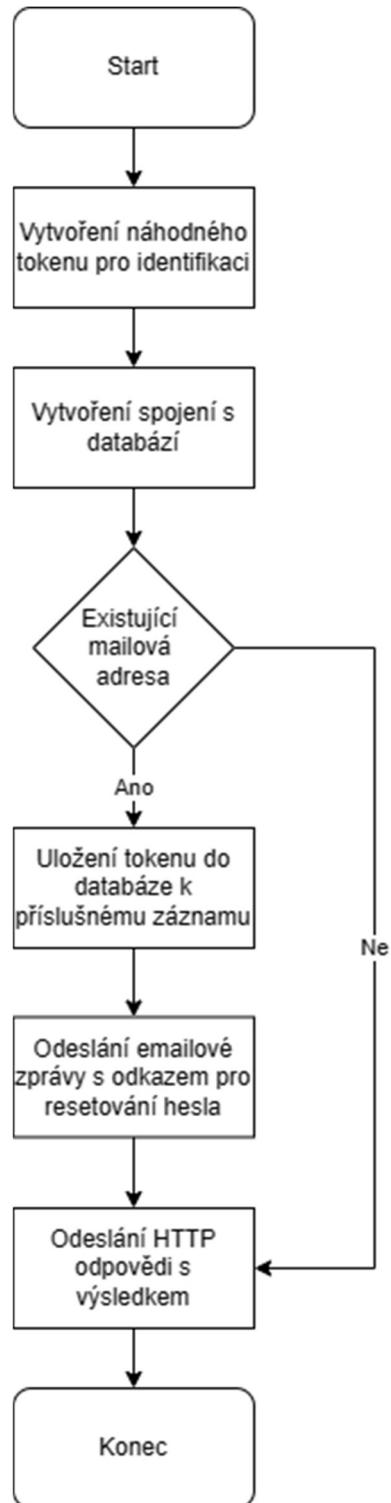
Příloha B.2.a: rcv_data



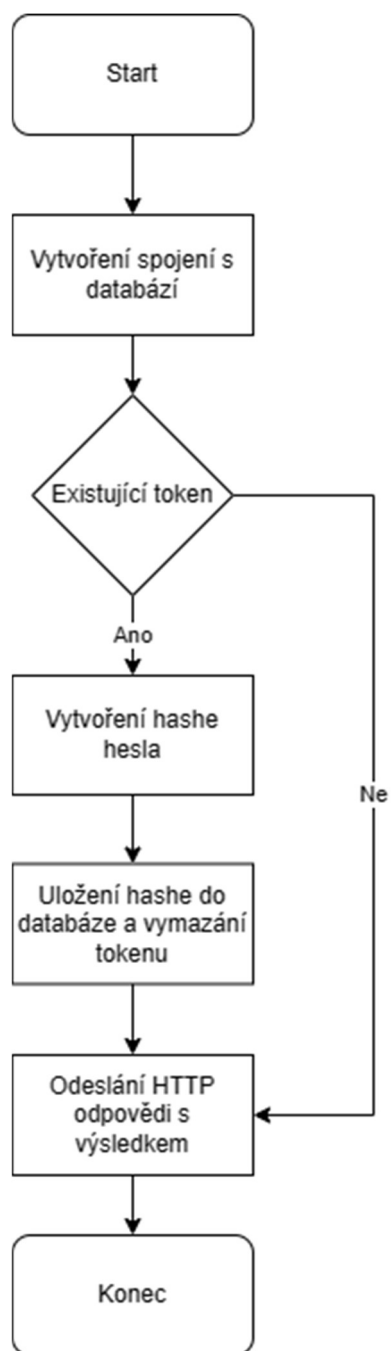
Příloha B.2.b: verify



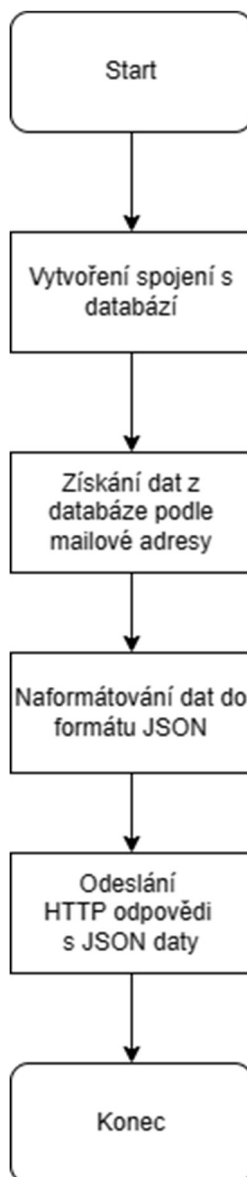
Příloha B.2.c: token



Příloha B.2.d: spass



Příloha B.2.e: cont



Příloha B.2.f: mreg



Příloha B.2.g: chpass1



Příloha B.2.h: setup1



Příloha B.2.i: setup2



Příloha B.2.j: setup3



Příloha C: Seznam obsahu tabulek v databázi

Tabulka spbox_tab

ID	INT unsigned
PD	TINYINT unsigned
TIME1	TIME
TIME2	TIME
TIME3	TIME
MO1	BIT
MO2	BIT
MO3	BIT
TU1	BIT
TU2	BIT
TU3	BIT
WE1	BIT
WE2	BIT
WE3	BIT
TH1	BIT
TH2	BIT
TH3	BIT
FR1	BIT
FR2	BIT
FR3	BIT
SA1	BIT
SA2	BIT
SA3	BIT
SU1	BIT
SU2	BIT
SU3	BIT
BAT	TINYINT unsigned
TEL	TINYTEXT <i>NULL</i>
INF	TEXT <i>NULL</i>
EMAIL	TINYTEXT <i>NULL</i>
IMSI	TINYTEXT <i>NULL</i>
SURL	TEXT <i>NULL</i>

Tabulka account_tab

id	INT <i>A/</i>
email	TINYTEXT
passw	TEXT
token	TEXT <i>NULL</i>