



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

OCHRANA BATERIE PRO ROBOT KAMBOT

APU UNIT FOR KAMBOT ROBOT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Aleksei Duz

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. František Burian, Ph.D.

BRNO 2018



Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Aleksei Duz

ID: 174283

Ročník: 3

Akademický rok: 2017/18

NÁZEV TÉMATU:

Ochrana baterie pro robot KAMBot

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s obvody zajišťujícími ochranu baterií Lilon a jejich balancováním.
2. Navrhněte a realizujte desku s plošnými spoji realizující základní ochranu 4 článkové baterie Lilon.
3. Plošný spoj by měl umožňovat zátěž zapnout stiskem tlačítka a vypnout zátěž dlouhým stiskem tlačítka.
4. Naprogramujte do řídicího čipu základní kód realizující ochranu baterie (zkrat zátěže, podpětí článku, přebití článku).
5. Otestujte parametry ochrany (podpětí, zkrat) a vyhodnoťte úroveň ochrany baterie.

DOPORUČENÁ LITERATURA:

CROMPTON, T. R. Battery reference book. 3rd ed. Boston: Newnes, 2000. ISBN 978-0-7506-4625-3.

POP, Valer. Battery management systems: accurate state-of-charge indication for battery powered applications. Dordrecht: Springer, 2008. ISBN 978-1402069444.

Termín zadání: 5.2.2018

Termín odevzdání: 21.5.2018

Vedoucí práce: Ing. František Burian, Ph.D.

Konzultant:

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato práce se zabývá návrhem a realizací PCB jednotky, která bude zajišťovat ochranu Li-Ion baterie pro robota KAMbot. Cílem práce je navrhnout, oživit a naprogramovat PCB desku, která se bude připojovat k 4-článkové Li-Ion baterie a zajišťovat její ochranu.

Práce je rozdělena na 4 kapitoly. První je věnovaná rozebrání různých způsobu ochrany baterii. Druhá kapitola se zabývá návrhem a realizace PCB desky pro ochranu Li-Ion baterii. Třetí část se věnuje programování mikrokontroléru, komunikaci s čipem ochrany baterii a ovládání vytvořené ochranné jednotky. V čtvrté kapitole jsou uvedené výsledky testu funkčnosti ochranného modulu Li-Ion baterii.

Klíčové slova

Ochrana Li-Ion baterie, návrh PCB, programování mikrokontroléru, čip kontroly stavu baterii

Abstract

This thesis deals with design and creation PCB unit, which will provide protection Li-Ion battery for KAMbot robot. The aim of this work is to design, revive and programm PCB board, whitch will be connected to 4-cells Li-Ion battery and will protect it.

The work is divided into 4 parts. The first one deals to parse different ways to protect the battery. The second part occupy design and creation PCB board for protection of aLi-Ion battery. The third section deals to programm microcontroller, communicate with chip that it protects battery and control the protection unit. The fourth part presents results of efficiency tests for protection Li-Ion battery unit.

Keywords

Protection Li-Ion battery, design PCB, programming microcontroller, checking chip of battery status

DUZ, A. Ochrana baterie pro robot KAMBot. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 60 s. Vedoucí bakalářské práce Ing. František Burian, Ph.D..

Prohlášení

”Prohlašuji, že svou závěrečnou práci na téma ochrana baterii pro robota KAMbot jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestně právních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.”

V Brně dne Podpis autora

Poděkování

Tímto děkuji svému vedoucímu bakalářské práce Ing. Františkovi Burianovi, Ph.D. za účinnou metodickou a odbornou pomoc a další cenné rady a připomínky při zpracování mé bakalářské práce a Ing. Tomašovi Lazně za cenné rady ohledně programování.

V Brně dne Podpis autora

Obsah

1	Úvod	13
2	Teorie	14
2.1	Li-Ion baterie	14
2.1.1	Historie	14
2.1.2	Ochrana a vlastností	14
2.1.3	Napětí a používání Li-Ion baterií	16
2.2	Nabíjení Li-Ion baterie	17
2.3	Balancování	18
2.3.1	Pasivní balancování	19
2.3.2	Aktivní balancování	24
2.3.3	Sjednocení balancování	25
2.4	Návrh PCB desky	26
3	Hardware	27
3.1	Návrh balanceru	27
3.1.1	MCU	28
3.1.2	RTC	30
3.1.3	Flash Paměť	31
3.1.4	Digital Isolator	32
3.1.5	Balancovací Integrovaný Obvod	33
3.1.6	Balancovací Obvod	34
3.2	Přípomínky k PCB desce	37
4	Software	40
4.1	Použitý software a pomůcky	40
4.1.1	IDE vývojové prostředí	41
4.1.2	STM32Cube	41
4.1.3	ST-Link Utility a programátor ST-Link V2	42
4.1.4	STMStudio	44
4.2	Programování	45
4.2.1	MCU	45
4.2.2	Balancovací čip	48
4.3	Rozbor programu	51
4.3.1	Inicializace	51
4.3.2	Nastavení registrů BQ7692006PW	53
4.3.3	Zpětná komunikace	54

5	Testování	56
6	Závěr	58
7	Literatura	59
8	Přílohy	60

Seznam obrázků

2.1	PCB control desky z různých válcových baterek	15
2.2	Příklad nabíječky s implementovaným balancerem	15
2.3	Napěťové rozsahy Li-Ion baterií[8]	17
2.4	Graf cyklu nabíjení Li-Ion baterie[3]	18
2.5	Jednoduchý omezovač napětí pro nabíjení Li-Ion[5]	20
2.6	Zapojení balanceru s balancovacím čipem BQ77PL900[6]	20
2.7	Princip funkčnosti čipu BQ77PL900[4]	21
2.8	Výsledek pasivního balancování na základě napětí[4]	21
2.9	Výsledek pasivního balancování na základě náboje[4]	22
2.10	Pasivní balancování článků s vnějšími FET tranzistory[4]	23
2.11	Nová topologie pro pasivní balancování článků s vnějšími FET tranzistory[4]	23
2.12	Technologie PowerPump[6]	24
2.13	Vyvažování článků pomocí technologie PowerPump[4]	26
3.1	Obecná bloková schémata zapojení balancovacího prvku	27
3.2	Bloková schémata zapojení balanceru	28
3.3	Zapojení MCU	29
3.4	Zapojení RTC	31
3.5	Zapojení Flash paměti	32
3.6	Zapojení I2C izolátoru	32
3.7	Zapojení balancovacího čipu a obvodu	34
3.8	Zapojení balancovacího obvodu	35
3.9	Top vrstva balanceru	38
3.10	Top vrstva balanceru	38
3.11	Zapájený balancer	39
3.12	Schéma virtuální baterie se zdrojem	39
4.1	Ukázka grafického prostředí STM32Cube	42
4.2	Nastavení clock konfigurace v STM32Cube	42
4.3	Uživatelské rozhraní programu ST-Link Utility	43
4.4	Uživatelské rozhraní programu STMStudio	44
4.5	Komunikace po I2C sběrnice	47
4.6	Přivedení clock signálu GPIO	51
4.7	Inicializace I2C sběrnice	52
4.8	Inicializace proměnných a bufferu	52
4.9	Nastavení hodinového signálu systému	53
4.10	Sekvence příkazů pro nastavení registrů BQ7692006PW	53

4.11	SYS_CTRL2 registr	54
4.12	Vyčítání SYS_STAT registru čipu BQ7692006PW	55

Seznam tabulek

3.1	Výkony rezistorů	36
3.2	Napětí příslušné proudům rezistorů	36
4.1	Připojení programátoru k balanceru	43

Zkratky

Li-Ion = Lithium-Iontová baterie

I2C = Inter-Integrated Circuit

MCU = Micro Control Unit

RTC = Real Time Clock

NC = Not Connected

IC = Internally Connected

LDO = Low-DropOut

IDE = Integrated Development Environment

HSI = High Speed Internal

HSE = High Speed External

PLL = Phase Locked Loop

GPIO = General Purpose Input Output

PCB = DPS = Deska Plošných Spojů

Kapitola 1

Úvod

V dnešní době svět se zabývá problémy okolního prostředí, technologií zachování energie a možnostmi použití elektrické energie jako alternativního zdroje paliva pro auta atd.

Díky tomu se vyvíjí spousta technologií v oblastí sbírání a zachování elektrické energie a tato oblast začíná být populární. Docela velké kroky v této oblasti dělá americká automobilka Tesla, která ukázala světu elektromobily s použitím Li-Ion baterky dojíždějící cca 650km na jedné baterie. To znamená, že vývoj Li-Ion a Li-Pol baterie je dnes nejperspektivnější oblast, protože oproti olověné baterie Li-Ion mají vysoké napětí, nízké samovybitení a výhodný poměr váhy a kapacity.

Je známe, že složitější baterie se skládají z několika sériově zapojených článků a výstupní napětí se rovna součtu napětí těchto článků. V ideálním případě každá část takové baterky je stejná a nabíjet nebo vybíjet jí můžeme jako jediný celek.

Bohužel, v reálném světě to tak nefunguje. Každý článek baterie je unikátní a najít aspoň dva stejných skoro není možné. Takže jejich charakteristiky s časem a teplotou se budou měnit(kapacita, napětí, ztráty atd.). Výrobce bateriek se snaží sestavovat baterky z článků, které mají praktické stejné parametry, ale rozdíly tam jsou. S časem tyto rozdíly charakteristik se budou zvětšovat a mohou přivést k destabilizace baterie. Z tohoto důvodu došlo k vývoji balanceru pro kontrolu a ovládání každého článků více článkové Li-Ion baterie.[5]

Cílem této práce je navrhnout, naprogramovat základní funkce, oživit a otestovat PCB desku pro kontrolu a ovládání nabíjení a vybíjení čtyř článkové Li-Ion baterie robota KAMbot.

Kapitola 2

Teorie

2.1 Li-Ion baterie

2.1.1 Historie

Vývoj Li a Li-Ion baterie se začal v roce 1912 skupinou vědců pod kontrolou G.N.Lewisa. V roce 1970 byli vytvořené první Li baterie, které byly nebezpečné za použití Li elektrod a v roce 1991 firma Sony začala vyrábět Li-Ion baterie, které používali místo nestabilních lithiových elektrod jenom ionty lithia.

Pro první Li baterie záporné elektrody byly vyrobené z koku a v roce 1997 koks byl vyměněn grafitem, díky čemu se zlepšila charakteristika napětí-vybíjení, zvětšil se proud, zmenšilo se ohřátí a baterie měly menší hodnotu samovybíjení. Kladné elektrody byly vytvořené z lithia a kobaltu nebo z lithia a manganu. První slitina měla velkou životnost, druhá byla bezpečnější.[3]

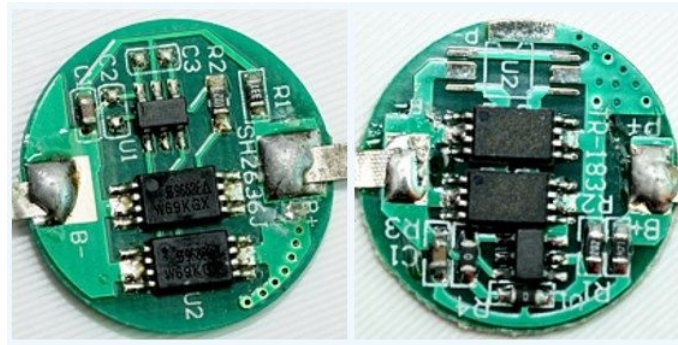
V dnešní době anoda se vyrábí z uhlíku, katoda je oxid kovu a elektrolyt představuje lithiová sůl v organickém rozpouštědle.

Li-Ion baterie jsou baterie běžně používané ve spotřební elektronice. V současné době je to v této oblasti asi nejpoužívanější typ, protože ji se dá snadno nabít a baterie může dosahovat malých rozměrů s dostatečně vysokou kapacitou a vysokým napětím. Li-Ion baterie mají nejlepší ochranu mezi ostatními typy bateriek. Je to z toho důvodu, že jsou celodenně v provozu (připojené k různým zařízením) a nacházejí se v rukách spotřebitele a jsou velice náchylné na destabilizaci při špatném chování s nimi a špatném způsobu nabíjení, proto se kladnou určitě požadavky na jejich ochranu.[3]

2.1.2 Ochrana a vlastnosti

Je už dávno známo, že Li-Ion baterie nemají rádi přebíjení, protože díky tomu oni ztrácejí svou kapacitu a zvětšují vnitřní odpor. Aby se baterka byla proti tomu chráněna, uvnitř Li-Ion baterie je čip, který kontroluje stav baterie. Někdy takové schémata se nazývají ochranné PCB desky. Tyto PCB hlídají hodnotu napětí a chrání baterie před přebíjením a podpětím, navíc hlídají teplotu a tlak uvnitř krytu

baterky. Obrázek 2.1 představuje 2 různé ochranné PCB, které se nacházejí v Li-Ion baterkách (válcové akumulátory).



Obrázek 2.1: PCB control desky z různých válcových baterek

Ochrana PCB je propojena s článkem baterie, sleduje a kontroluje ho. Je už jasné, že ty desky sleduje kritické stavy baterie. Většinou tato ochrana obsahuje tranzistor, který při dosáhnutí napětí 4,1V až 4,3V se otevře a zkratuje napájení, tím se zastaví nabíjení. Pro přerušení nabíjení při hodnotě napětí od 4,1V do 4,2V dnešním bateriím nic nehrozí, v prvním případě nedobijí několik procent nebo desetin procent a v druhém případě bude nabita naplno. Ale v situaci odpojení baterky při hodnotě napětí 4,3V, článek se bude systematické přebíjet, díky čemu se sníží kapacita a zkrátí se doba života baterie. Bohužel taky je možnost, že článek může nabobtnat. Pak začíná hrozit nebezpečí výbuchu baterky, proto existuje v některých bateriích tlaková ochrana. A ta dělá to, že zkontroluje tlak uvnitř krabičky baterky a při hodnotě 1034kPa rozpojí spojení baterii a zátěži. Taky existuje schéma ochrany i proti velkému vybití, která sleduje napětí baterky a při vybití v rozmezí hodnot 2,3V až 2,5V odpojí článek od zátěži. A posledním zabezpečením desky je termální ochrana proti přehřátí a to funguje tak, že, jakmile článek dosáhne napětí 90C, od baterky se odpojí zátěž. Ale ty ochranné PCB desky Li-Ion články občas nemají.[3]



Obrázek 2.2: Příklad nabíječky s implementovaným balancerem

Pro ochranu všech článků je potřeba mít speciální PCB desky (balancery), které limitují nabíjecí nebo vybíjecí proudy a zachrání baterie před přebitím nebo podbitím. V dnešní době pro složitější elektrické spotřebiče se používají několika článkové baterie. Aby takové baterky vydrželi co nejvíce, nabíjení se musí provádět správným způsobem a to za použití balanceru. Inteligentní nabíječka, která je zobrazena na obrázku 2.2, má integrovanou balancovací PCB desku, napojí se na baterie a sleduje nabíjení resp. vybíjení každého článku zvlášť.

Li-Ion baterie mají schopnost stárnout a to je dáno tím, že v některých látkách těchto baterií probíhají nevratitelné chemické reakce, které svádějí baterku do nefunkčního stavu.[3]

Výhody Li-Ion baterie jsou následující:

- vysoké napětí
- malé samovybití
- žádný paměťový jev
- jednoduše nabíjení
- nevyžaduje údržbu
- výhodný poměr váha / kapacita[7]

Pro správné pochopení výhod, je třeba vysvětlit paměťový efekt. Tento jev je vztažen k NiCd baterie a vzniká při opakovaném vybíjení na stejnou hloubku vybití. Paměťový efekt je náhlý pokles napětí článku zhruba o hodnotu 50 až 100mV, přičemž nedochází k poklesu kapacity. Podstatou jevu je změna krystalové struktura záporné elektrody a ten efekt lze odstranit plným vybitím baterie.[10]

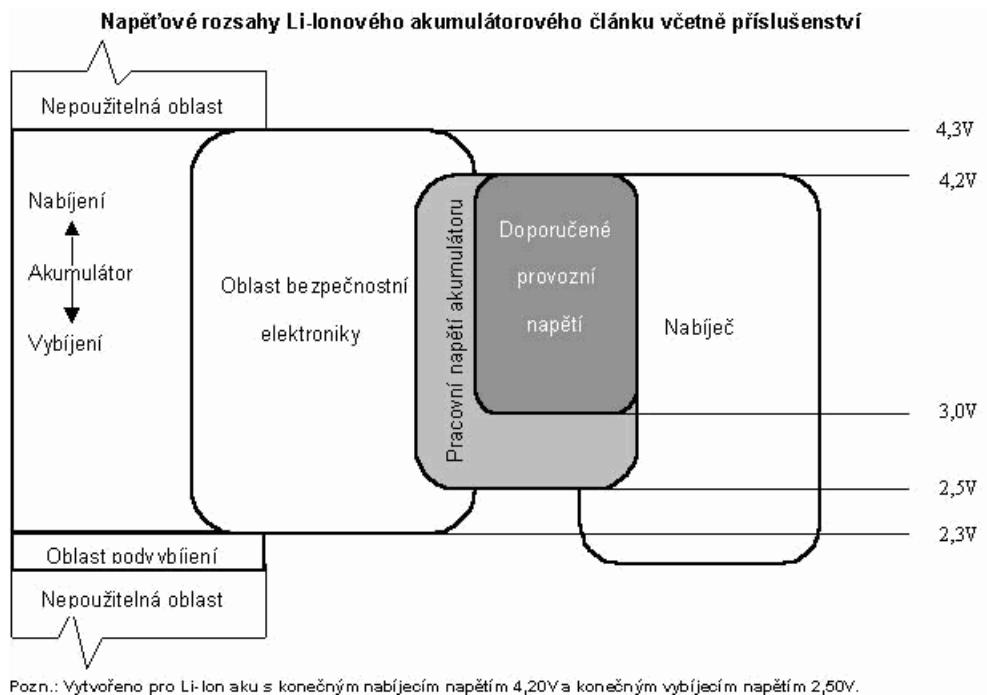
Nevýhody Li-Ion baterie jsou:

- nutnost ochranné PCB desky
- rychlé stárnutí
- špatně pracuje při nižších teplotách
- velký vnitřní odpor[7]

2.1.3 Napětí a používání Li-Ion baterií

Je nutno při práci s Li-Ion baterkami rozlišovat skutečné a jmenovité napětí. Jmenovité napětí článku je stanoveno normou a uvádí se na nálepce baterie, toto napětí se přibližně rovná průměrnému napětí při vybíjení za obvyklých provozních podmínek. Při této hodnotě články by se měly odpojit. Bohužel není tato hodnota jednoznačně stanovena a závisí na použité státní normě. Skutečné napětí je hodnota, která byla naměřena na svorkách baterie a závisí na teplotě, stavu nabití, na směru a velikosti protékajícího proudu a na několika dalších faktorech. Je nutno ještě upozornit, že u některých bateriek výrobce uvádí maximální hodnotu nabíjecího napětí, která nesmí být za žádných okolností překročena.[9]

Na obrázku 2.3 je vidět napěťové rozsahy pro Li-Ion baterie. Bohužel tento obrázek neznázorňuje hodnoty napětí, které byly stanoveny normami. Jak již bylo řečeno, jmenovité napětí baterie je stanoveno státní normou, proto napětí odpojení baterie



Obrázek 2.3: Napětové rozsahy Li-Ion baterií[8]

se nerovná 3V, ale je vyšší. Sice je možnost vybití baterie i pod stanovené normy, ale baterie bude používat obyčejný uživatel, proto se mají dodržovat normy. Například podle normy v USA hodnota napětí odpojení článku je 3,7V a podle normy stanovenou EU jmenovité napětí se rovná 3,6V. Takže pro čtyřčlávkovou baterie odpojovací napětí se bude rovnat 14,4V a napětí plného nabití bude 16,8V.

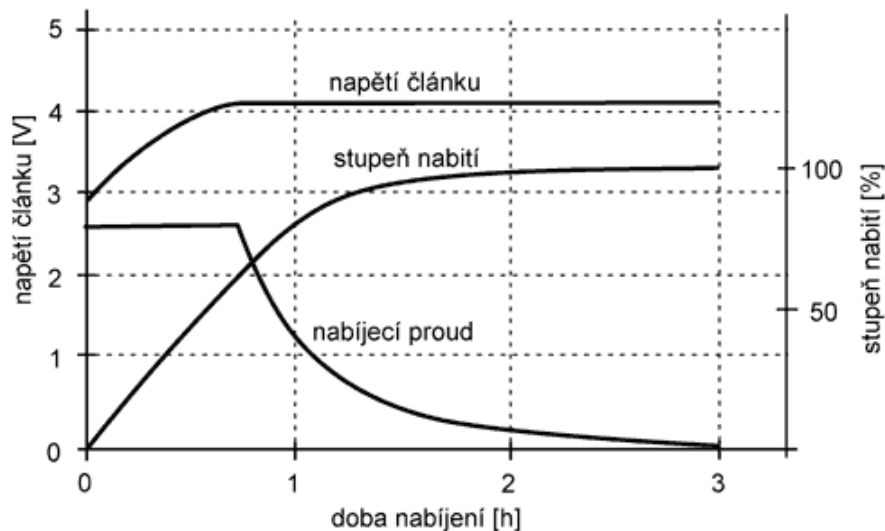
2.2 Nabíjení Li-Ion baterie

Li-Ion baterie mají docela přesné požadavky při nabíjení a vybíjení, a jsou to přesné hodnoty napětí, při kterých je potřeba odpojovat baterku od nabíječky nebo zátěži.

První Li-Ion baterie s použitím grafitu měli za potřebu odpojovat napětí při hodnotě 4,1V, i když použití vyššího napětí umožňovalo zvětšit energetickou pevnost. Bohužel nabíjení vyššími napětí zkracovalo dobu života baterky. Pak se začali objevovat Li-Ion baterie, které už měly hodnotu odpojení 4,2V díky přidání různých chemických přísad. Takže je potřeba si uvědomit, že není ta hodnota odpojení přesná, ona se pohybuje v rozmezí od 4,15V do 4,25V.

Li-Ion baterie budou nabity naplno, když dosáhnou napětí odpojení, a proud klesne na hodnotu 3% od začátečního nabíjecího proudu. Většinou nabíjení Li-Ion baterie trvá cca 2-3 hodiny při proudu 1C.[3]

Cyklus nabíjení baterie z obrázku 2.4 se dá rozdělit na dvě oblasti. První část grafu se dá pojmenovat jako nabíjení konstantním proudem, kde přes baterie teče maximální



Obrázek 2.4: Graf cyklu nabíjení Li-Ion baterie[3]

nabíjecí proud, pokud napětí na článku baterie nedosáhne napětí odpojení(4,2V). Druhá část by se pojmenovala jako nabíjení konstantním napětím, kde procházející proud baterkou klesá do hodnoty 3% od původní hodnoty.

Li-Ion baterie jsou bezpečné jen když se dodržují pravidla jejich použití(nabíjení a vybíjení). Jestli napětí baterky bude větší než 4,2V, baterie může fungovat nestabilně a časem přestane fungovat. Je to z důvodu toho, že při napětí nabíjení 4,3V v baterce probíhá metalizace anody lithiem a katoda aktivně produkuje kyslík díky tomu roste teplota baterii, proto je potřeba věnovat hodně pozorností jejich zabezpečení a správnému nabíjení či vybíjení.[3]

2.3 Balancování

Aby opotřebení baterie probíhalo nejpomaleji články mají být balancované, o tom se stará balancer. Balancer je ochranná PCB deska nebo zařízení, která se stará o více článkovou baterie tím, že kontroluje a ovládá nabíjení nebo vybíjení a limity napětí a proudy každého článku.

Existuje dva kritéria pro hodnocení úrovní balancování:

- vyrovnání napětí na člancích
- vyrovnání náboje na člancích

Při vyrovnání napětí články musí mít stejné napětí a při vyrovnání náboje se nějakým způsobem vypočítá náboj, porovná se s ostatními články a doplní se náboj tam, kde neodpovídá ostatním článkům(napětí se může lišit).

Dosáhnout svého cíle v balancování těmito metodami je možné následujícími způsoby:

- pasivní balancování
- aktivní balancování

Pasivní balancování funguje způsobem převodu zbytečné elektrické energii v tepelnou. Například jeden z článků baterie se nabil naplno a ostatní ještě potřebuje nějaký čas na dobití, na nabitém článku se zapne balancování. Nabíjecí proud tohoto článku klesne díky tomu, že se otevřela vyvažovací větev, kde se spali nadbytečný proud. Aktivní balancování funguje jiným způsobem, princip spočívá v tom, že náboj bude se transportovat z jednoho článku do druhého s nejmenšími ztrátami do té doby, pokud nebude vyrovnán náboj na člancích.

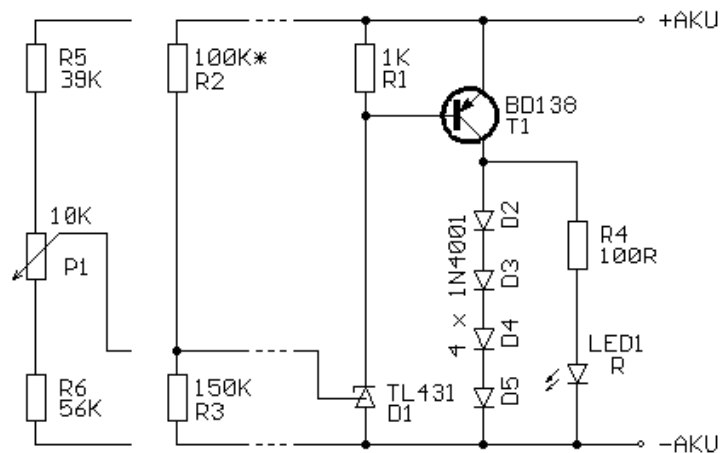
Prostým způsobem balancování je pasivní, protože porovnání napětí a vyzařování zbytečné energii je jednodušší než porovnávat náboje v člancích a transportování zbytečného náboje tam, kde je jejich nedostatek. Většinou balancování se provádí při nabití baterky, když energii je dost a může se nešetřit, proto se často používá pasivní balancování. Na druhou stranu se dá balancovat aj vybíjení baterie, ale v tomto případě se objevuje velice důležitý bod balancování a to je šetření energii, protože ta energie se bere z baterky a nesmí se vyzařovat jako teplo, proto při balancování vybity se používá aktivní balancování. Bohužel tyto schémata jsou výjimečné, protože jsou velice složité.[5]

2.3.1 Pasivní balancování

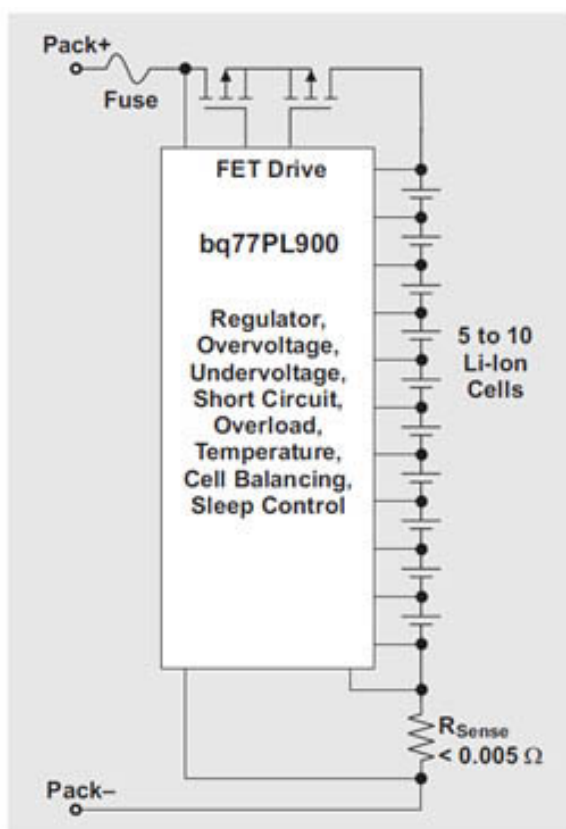
Nejjednodušší způsob balancování je pasivní(omezovač napětí). To funguje tak, že se porovnává napětí na článků s limitním napětím 4,2V a při dosažení hodnoty tohoto napětí, otevře se výkonový tranzistor, který je připojen paralelně k článku a začne propouštět velkou část proudu a transformovat elektrickou energii v tepelnou. Během toho články dostávají malou část proudu, což skoro zastavuje nabíjení článku a dává možnost se dobit ostatním. Fakticky k vyrovnání napětí při takovém způsobu balancování dochází na konci nabíjecí sekvenci při dosažení limitní hodnoty napětí 4,2V.

Na obrázku 2.5 je představen jednoduchý balancer bez použití speciálních čipů a s použitím výkonového tranzistoru BD138. Cílem toho zapojení je nabíjení a vyrovnání napětí na člancích. Na tomto obrázku jsou představeny dvě varianty pro nastavení napětí. První varianta má nastavené napětí na pevnou, druhá varianta má možnost nastavení napětí pomocí trimru. Takové balancery jsou stanoveny ručně doma a potřebují dobré chlazení k vykonání své činnosti, potřebují radiátor. Oproti tomu všechny firemní balancery jsou staveny s použitím mikrokontrolérů a fungují jinak.[5]

Na obrázku 2.6 je představen čip BQ77PL900, který funguje tak, že porovnává napětí článků s naprogramovanými limitami a podle potřeby zapíná režim balancování. Na obrázku 2.7 je vidět, jak funguje nabíjení. Jestli napětí jakéhokoliv článků přeskakuje nastavený horní limit, nabíjení toho článku se zastaví a bude vypnuto do té doby,



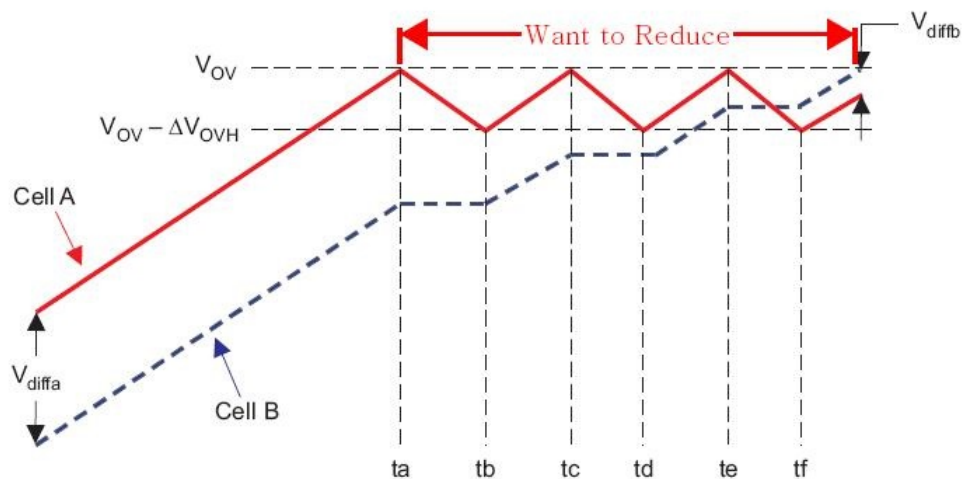
Obrázek 2.5: Jednoduchý omezovač napětí pro nabíjení Li-Ion[5]



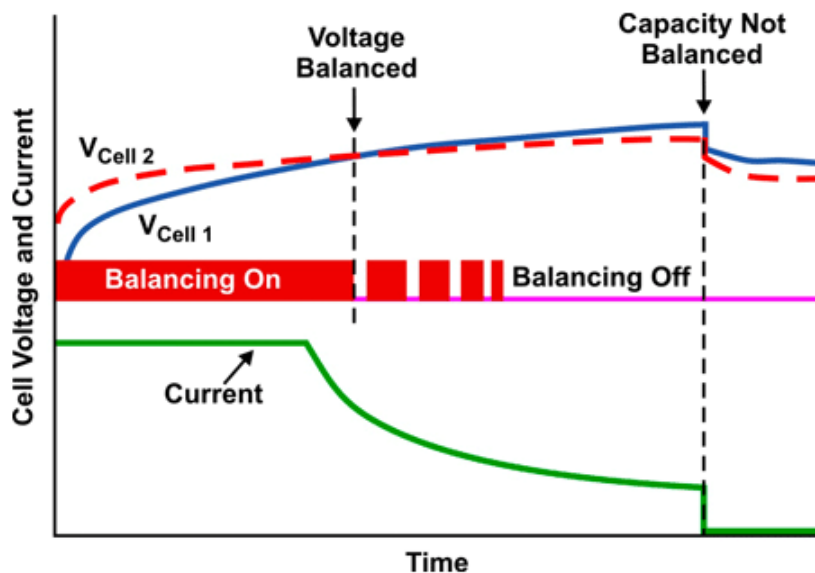
Obrázek 2.6: Zapojení balanceru s balancovacím čipem BQ77PL900[6]

pokud napětí neklesne pod nastavený dolní limit nabití. Pak jestli balancování ještě není dokončeno a článek se trochu vybil, tak nabíjení se zapne zase a bude se to opakovat, pokud všechny články nebudou v rozmezí nabití.[6]

Při použití tohoto algoritmu balancování se používá jen informace o napětí, díky čemuž existuje možnost, že balancování nebude na 100% dokončeno. Je to z důvodu toho, že články mají různou impedanci a to má vliv na napětí při nabíjení. Čip není schopen poznat, jaký je důvod rozbalancování napětí. Z jedné strany to může být rozdílem kapacity, z druhé strany je to způsobeno rozdílem vnitřních impedancí. Proto není zajištěno, že každý článek bude nabit na žádanou hodnotu 100% kapacity. Obrázek 2.7 ukazuje výsledek takového způsobu balancování a vliv rozdílu impedancí, o kterém bylo již dříve řečeno, je zahrnut v napěťovém rozdílu V_{DIFFA} a V_{DIFFB} . [4]



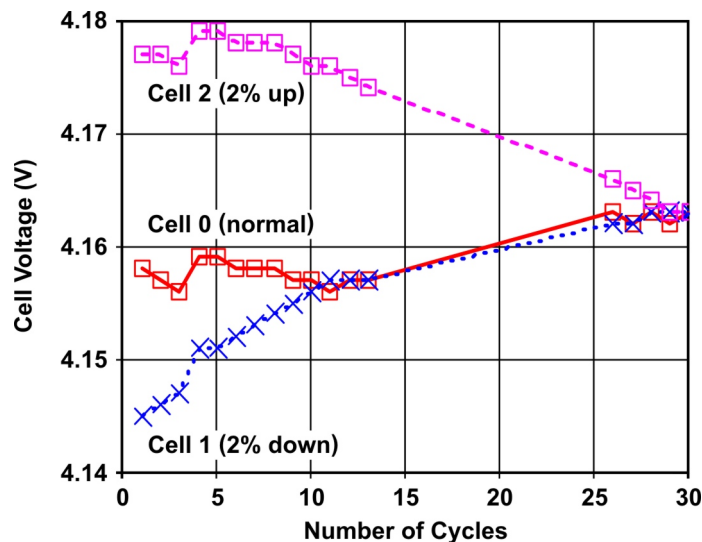
Obrázek 2.7: Princip funkčnosti čipu BQ77PL900 [4]



Obrázek 2.8: Výsledek pasivního balancování na základě napětí [4]

Oproti čipu BQ77PL900 čip BQ2084 provádí balancování blíž ke konci nabíjení, když hodnota nabíjecího proudu není velká. To se dělá z důvodu toho, aby se minimalizoval vliv vnitřních impedancí článků. Další výhodou BQ2084 je to, že měří, analyzuje napětí na všech článcích a rozhoduje o vyvažování článků. Výsledkem toho je efektivnější balancování. Tato výhoda se může používat jen v režimu nabíjení.[6]

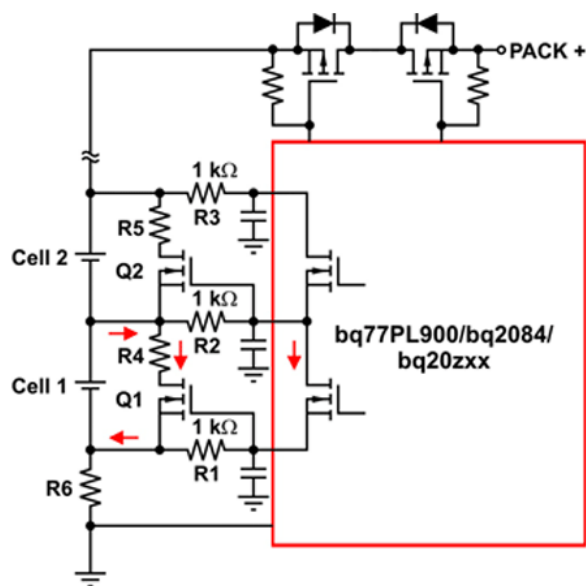
Oproti těmto čipům (BQ77PL900 a BQ2084) čipy rodiny BQ20ZXX používají jiný kritérium balancování než porovnání napětí a to je porovnání náboje na článcích. Oni fungují na technologii Impedance Track a vypočítávají stav nabití a kapacitu. Jinak řečeno, tato technologie pro každý článek vypočítá náboj Q_{NEED} , který je potřebný pro dosažení plné nabití stavu. Dále najde rozdíl ΔQ mezi Q_{NEED} všech článků a stanoví nulovou hodnotu pro ΔQ . V následujícím kroku čip začíná balancování a to takovým způsobem, že použité algoritmy spíná vyvažovací tranzistory FET během nabíjení článků tak, aby byl dodán potřebný náboj Q_{NEED} . Cílem je svest rozdíl ΔQ k 0 pro každý článek. Velkou výhodou je to, že vyvažování není zkresleno impedančními rozdíly jednotlivých článků. Pracuje tato metoda nezávisle na tom, zda je baterie nabíjena, vybita nebo je dokonce mimo provoz. Při použití technologii Impedance Track lze dosáhnout lepšího balancování.[4][6]



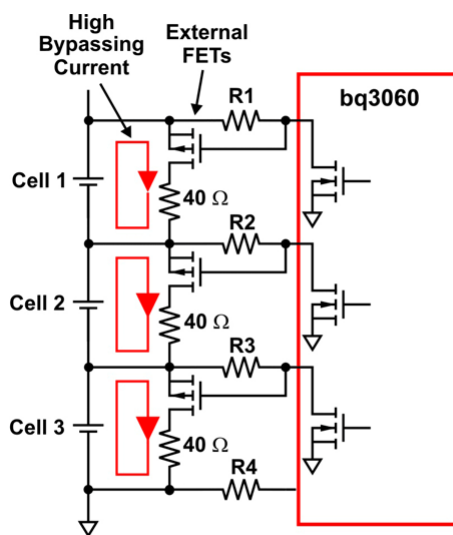
Obrázek 2.9: Výsledek pasivního balancování na základě náboje[4]

Protože pasivní balancování s integrovaným tranzistorem FET má některá omezení, lze při určitých odchylkách z titulu vnitřní nerovnováhy článků pochybovat o účinné a dostatečné formě vyvážení. Bez ohledu na to, v důsledku malého balancovacího proudu, může korekce pro typickou velikost nevyvážení trvat i několik cyklů. Dá se tuto situaci zlepšit tím, že k čipu se doplní vnější balancovací obvod.[4]

Zapojení na obrázku 2.10 funguje tak, že nejdřív dochází k otevření vnitřního FETu, pak začíná přes rezistory vnějšího obvodu, které jsou připojeny k článkům a vývodům integrovaného obvodu, protékat malý proud a díky tomu, pak se spíná vnější



Obrázek 2.10: Pasivní balancování článků s vnějšími FET tranzistory[4]



Obrázek 2.11: Nová topologie pro pasivní balancování článků s vnějšími FET tranzistory[4]

FET tranzistor. Má to jednu nevýhodu a to je to, že sousedící články nemohou být rychle a současně vyvažovány. Například pokud bude sepnut sousední vnitřní tranzistor, přes R2 nebude protékat žádný proud, a proto nelze aktivovat tranzistor Q2.

Zapojení na obrázku 2.11 funguje stejným způsobem jako předchozí, ale jediný rozdíl je v tom, že tento čip nemá vnitřní balancování článku. Protože je ale integrovaný

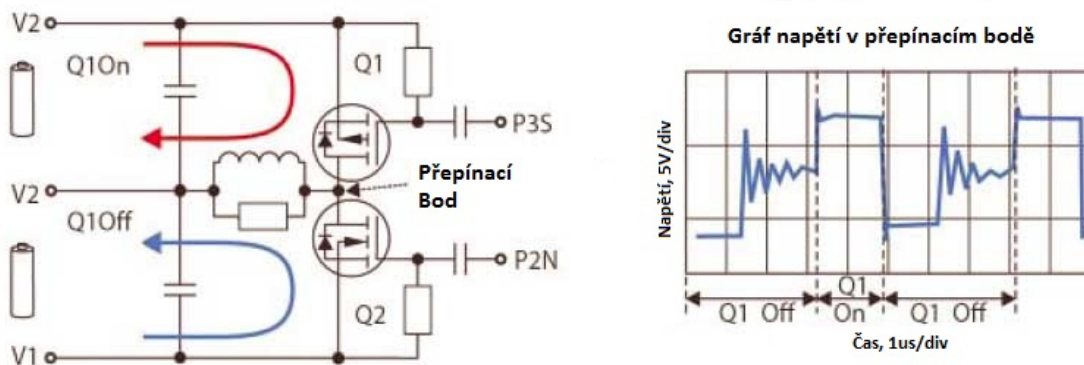
spínací prvek pro vyvažování k dispozici s otevřeným kolektorem, může současně obsluhovat několik článků bez ohledu na jejich pozici, buď to sousední články nebo ne, což umožňuje poskytovat efektivnější vyvažování. Struktura čipů na obrázcích 2.10 a 2.11 využívají algoritmu balancování v závislosti na napětí.[4]

2.3.2 Aktivní balancování

Z hlediska efektivity aktivní balancování je lepší než pasivní, protože pro přenášení energie od více nabitého článku k méně nabitému místo rezistorů se používá kapacitní nebo induktivní přenos energie. Energie je směrována tam, kam je skutečně potřeba a nebude bez užitečně protopena na rezistorech nebo výkonových tranzistorech, proto ztráty energii prakticky v tomto vyvažování nejsou. To se používá v aplikacích, kde je potřeba maximálně zvětšit dobu využití baterii bez dobíjení.[4]

Čip BQ78PL114 firmy Texas Instruments byl vyvinut speciálně pro aktivní vyvažování a používá induktivní přenos pro transportování energie. Tento integrovaný obvod používá technologii PowerPump. Tato technologie používá pro přenos náboje mezi sousedními články dva MOSFET tranzistory s kanály P a N a cívku. Na obrázku 2.12 je představen zjednodušený obvod typu boost-buck technologií PowerPump.[6]

Vývojář definuje úroveň rozbalancování mezi sériově zapojenými články baterie a jakmile čip detekuje nevyvážení nad touto prahovou úrovní, dochází k aktivaci přenosu energie mezi články. Například integrovaný obvod stanovil, že je potřeba udělat přenos energie z horního článku V3 do dolního článku V2. Na výstupu P3S se objeví signál s kmitočtem 200kHz a střídou 30%. Tranzistor Q1 se otevře a spustí přenos energie do cívky. Jakmile signál P3S se vypne, tranzistor Q1 se zavře a energie v cívce dosáhne své maximální hodnoty, aby přenos se dokončil je potřeba uzavřít proudovou smyčku, s tím pomůže substrátová dioda tranzistoru Q2, která v tuto chvíli je průchozí a dovršuje tak přenos náboje do článku V2. Ztráty při tomto přenosu nejsou velké a objevují se v diodě a cívce, která má malý sériový odpor.[4]



Obrázek 2.12: Technologie PowerPump[6]

Výhodou technologie PowerPump je to, že žádaného stavu vyvažování lze dosáhnout bez ohledu na napětí jednotlivých článků. Navíc přenos náboje se provede v průběhu jakéhokoliv režimu, buď to nabíjení, vybíjení nebo klidový stav. Dokonce k přenosu náboje dojde i v případě, když článek, který poskytuje náboj, bude mít nižší napětí než cílový článek.

BQ78PL114 nabízí 3 balancovací algoritmy:

- čerpání dle svorkového napětí
- čerpání dle napětí naprázdno
- čerpání dle stavu nabití

Čerpání dle svorkového napětí je podobné pasivnímu balancování a taky má problémy s vyvážeností kapacity v důsledku odlišných impedančních poměrů. Čerpání dle napětí naprázdno kompenzuje rozdíly impedancí odhadem napětí naprázdno na základě měření proudu sestavy a impedancí článků. Čerpání dle stavu nabití pracuje stejným způsobem jako integrované obvody s technologií Impedance Track, nabití každého článku a přenos energií mezi články se provádí na konci nabíjení. Obrázek 2.13 představuje graf vyvážení článků technologií PowerPump.

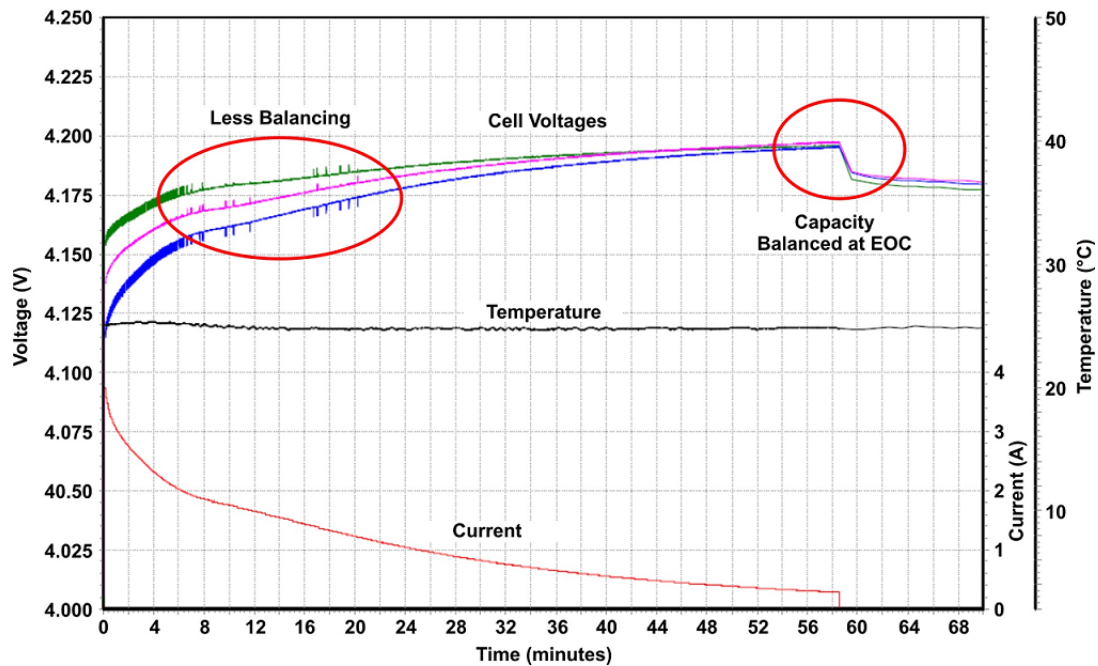
Technologie PowerPump je efektivnější než běžné pasivní balancování a umožňuje balancování v jakémkoliv režimu baterie. Takže změnou vnějších součástí může být jednoduše řízen i větší vyvažovací proud. Aktivní balancování má výhodu i pro bezpečnost. Ta výhoda se skrývá v možnosti sledovat čas potřebný k vyvážení jednoho článku a čisté čerpání energie pro každý prvek (počet čerpání do článku mínus počet čerpání z článku). Jestli tato hodnota bude moc vysoká, tedy článek dostává hodně energie. Tato informace pomůže bez problému odhalit vadný článek a být jednou z definic pro stanovení aktuální kondice baterie, stejně jako je tomu v případě impedance článku nebo jeho kapacity při plném nabití.[4]

2.3.3 Sjednocení balancování

Nové technologii a správné balancování je jednou z cest zvětšení doby použití baterie a díky tomu, že se kontroluje stav každého článku, zvětšují se i bezpečnost exploatace bateriek.

Při pasivním balancování cílem je jenom vyvážit články mezi sebou, ale z důvodu malých proudů to se může provádět jenom na konci nabíjecího cyklu. Přebití baterky nemůže nastat, ale to neztvětší dobu fungování bez nabíjení, protože hodně energii bude ztraceno ve vnějších balancovacích obvodech.

Při použití aktivního vyvážení pomocí technologii PowerPump zároveň se řeší dva cíle, první je balancování napětí a druhý je balancování náboje. Energie se zásobuje a odevzdává se nejslabšímu článku místo toho, aby se ztrácela jako teplo v balancovacích obvodech.



Obrázek 2.13: Vyvažování článků pomocí technologie PowerPump[4]

2.4 Návrh PCB desky

Před tím, jak se začne navrhovat a kreslit prototyp desky balanceru, návrhář musí:

- rozhodnout, co PCB má umět
- dohledat součástky a přečíst jejich datasheety a manuály
- rozhodnout, jaká bude použita komunikace (I2C, SPI atd.) nebo jestli vůbec bude potřeba ji mít
- pokusit se najít nějaké schéma Evaluation nebo Developer Boardu k vyhledávaným součástkám

Při návrhu PCB je dobré na začátku používat schémata referenčních zapojení součástek, které ve většině případů se nacházejí v datasheetech. Tam je vždy popsáno, jak například správně se má provádět napájení čipu nebo jak se správně zapojují piny. To se týče většinou nějakých speciálních pinů, které mají za funkci například nastavení režimu. Takže datasheet udává informace o tom, který vývod čipu, co znamená a jakou má funkci. Dokonce většina pinů jsou multifunkční.

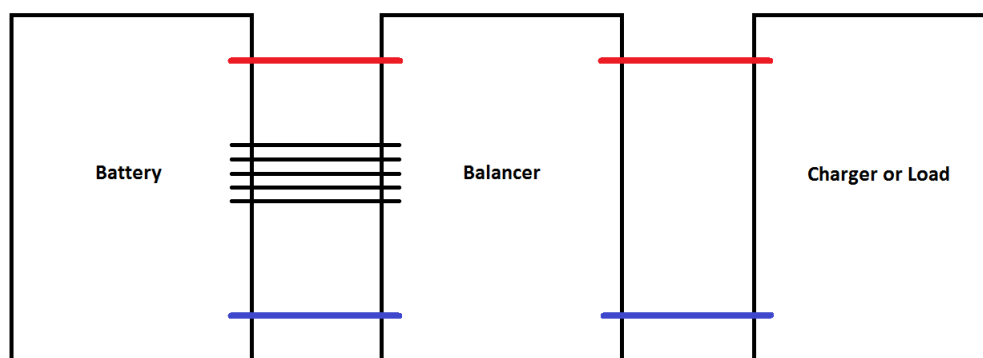
Někdy se stává situace, že pro vytvoření schématu referenčního zapojení z datasheetu nestačí. Například tam chybí parametry součástek. V těchto případech výrobce složitějších čipů na svých internetových stránkách mají speciální soubory, ve kterých ukazují schémata s zapojením daného čipu a soupisem použitých součástek a popisem jejich parametrů. Takové soubory se nazývají application notes.

Kapitola 3

Hardware

3.1 Návrh balanceru

PCB balancer je ochranná deska (ochranný prvek), která se přidává mezi baterii a nabíječkou či zátěží. Primární funkcí tohoto mezi kusu je hlídání napětí na článcích, vyvážení článků a odpojení zátěže či nabíčky při dosažení limitních hodnot. Obrázek 3.1 představuje obecné zapojení balanceru v systému robotu KAMbot nebo při nabíjení.

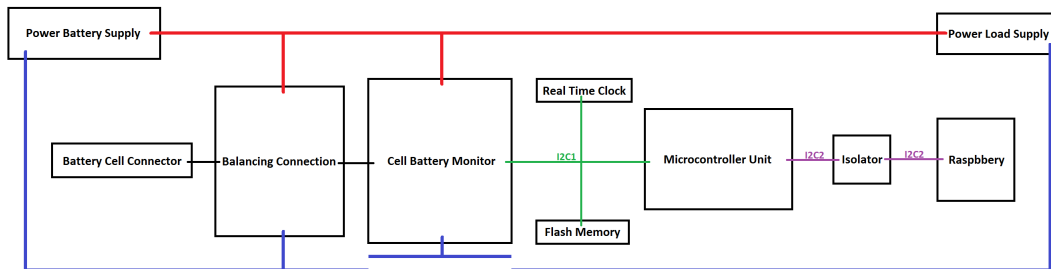


Obrázek 3.1: Obecná bloková schémata zapojení balancovacího prvku

Balancer bude obsahovat vyvažovací zapojení, balancovací čip kontrolující napětí na článcích, zapojení určené pro vyvážení, microcontroller, flash paměť, real-time clock a komunikační izolátor.

Tato bakalářská práce je zaměřená na vytvoření prototypu balanceru pro Li-Ion baterie. Tento prototyp bude použit na robotě KAMbot, který má jako řídicí systém implementovaný jednodeskový počítač Raspberry Pi 2. Hlavním cílem práce je výroba prvního prototypu, na kterém bude vyzkoušeno vyvažovací zapojení, otestován

balancovací čip, komunikace s ním a jeho ovládání. Takže je potřeba říct i o konečném cíle vývoje, netýká se to zadání bakalářské práci. Výsledkem bude bezpečnostní modul pro čtyř článkovou Li-Ion baterie, který bude kontrolovat nabíjení a vybíjení baterky, automatické odpojovat zátěž a řídicí obvod balanceru od baterie při dosažení určitého úrovně vybíjení. PCB bude obsahovat RTC, což umožní sledovat časy, a flash paměť, kam se bude ukládat čas a napětí článků při dokončení nabíjení nebo odpojení baterie od zátěži.



Obrázek 3.2: Bloková schémata zapojení balanceru

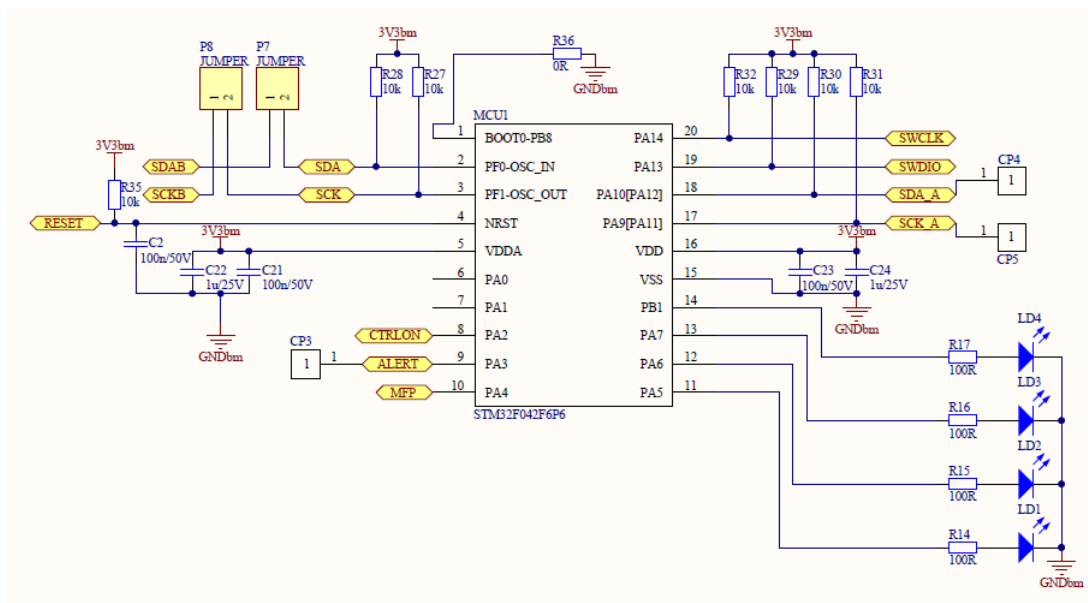
Obrázek 3.2 předvádí, jak vypadá bloková schéma PCB balanceru a prvního prototypu. Z obrázku je vidět, že na desce je dvě Inter-Integrated Circuit (I2C) komunikace. I2C1 je interní komunikace, která řeší přenosy dat uvnitř balanceru, a I2C2 je externí komunikace, která se stará o komunikaci s řídicím systémem robotu KAMbot.

3.1.1 MCU

Pro kontrolu a automatické ovládání balanceru zapojení musí obsahovat řídicí jednotku. Tady se používá MCU STM32F042F6P6 od firmy STMicroelectronics. Tento řídicí čip se stará o nastavení registru balancovacího čipu, získání z něho užitečných dat a jejich následném zpracování, nastavení RTC, ukládání dat do flash paměti, komunikaci s řídicím systémem robotu a vizuální indikaci pomocí LED diod.

Na obrázku 3.3 je uvedeno zapojení microcontrolleru. Tento řídicí čip je docela malý (footprint TSSOP20), což je výhodou, protože balancer musí být velice malým. MCU patří do řady ARM® 32-bit Cortex®-M0 CPU s rozměrem programovatelné vnitřní flash paměti 32 Kbytes a pracovním rozsahem teplot od -40 do 85 °C. Microcontroller má integrované RTC, vnitřní oscilátor na 8 MHz a 48 MHz s možností PLL násobiče respektive děliče frekvenci a jednu I2C komunikaci s maximální rychlostí 1 Mbit/s (Fast Mode Plus).

Aby se správně zapojil řídicí čip je potřeba přečíst jeho referenční manuál, ve kterém jsou popsány všechny registry, režimy, nastavení atd. Tím se dá rozhodnout a pochopit, jak je potřeba zapojit speciální piny odpovídající za určité funkce při



Obrázek 3.3: Zapojení MCU

nastartování procesoru. Je také nutno rozebrat zapojení každého pinu a je zásadně si pamatovat, že obyčejné piny mají více než jednu možnost funkčnosti.

Pin 1 je speciální pin a odpovídá při startování MCU za to, odkud procesor začne číst kód programu, z jaké adresy nebo místa. Existuje několik možností, buď to vnitřní flash paměť, systémová paměť nebo SRAM. Pro danou aplikaci je třeba, aby řídicí čip si spouštěl program, který bude nahrán do vnitřní flash paměti, proto díky tabulce režimů z referenčního manuálu (kapitola 2.5), pin 1 se připojuje na zem.

Piny 2 a 3 se používají pro připojení externího oscilátoru nebo pro komunikaci po I2C sběrnici. V této aplikaci přednost má komunikace. V I2C sběrnici pin 2 má funkci přenosu dat a pin 3 odpovídá za hodinový signál celé sběrnice. Vývody 2 a 3 tvoří interní I2C sběrnici balanceru. Aby tato I2C komunikace fungovala správně, musí být splněna důležitá podmínka a to je připojení pull-up rezistorů.

Pin 4 odpovídá za externí resetování MCU a je vyveden na programovací konektor. Tento vývod funguje tak, že jakmile na něm pádné úroveň napětí se provede resetování procesoru. Aby bylo zaručené, že microcontroller se nebude restartovat náhodně k tomuto pinu je připojen pull-up rezistor a filtrační kondenzátor pro vyhlazení signálu. Je nutno říct, že tento pin není zatížen integrovaným obvodem, který zajišťuje resetování MCU při stabilizaci napájecího napětí, aby se microcontroller spustil bez problému a nezkolaboval v neurčitěm stavu. Je to jen z důvodu toho, že dané MCU už má v sebe tuto funkci implementovanou (referenční manuál, podkapitola 6.1.1).

Piny 5, 15 a 16 jsou napájecí a zemní. Vývod 5 odpovídá za napájení analogové části MCU, pin 16 má za úkol napájet digitální část a vývod 15 je společná zem pro celý microcontroller. Napájení k přírodním pinům má připojené stabilizačně-

filtrační kondenzátory, které se starají o potlačení a vyrovnaní výkyvu napájecího napětí. Takže napájení a zem jsou propojené s programovacím konektorem.

Piny 19 a 20 jsou programovací piny. Pomocí nich MCU se propojuje s programátorem, který tvoří most mezi MCU a počítačem. Tyto dva piny stejně jako RESET pin a napájení s zemi jsou vyvedené na programovací konektor.

Piny 6 - 14, 17 a 18 jsou vývody s různými funkcemi, se kterými návrhář může zacházet libovolně podle předepsaných možností. Je třeba ještě upozornit, že vývody 17 a 18 spojují v sobě dva různé piny s odlišnými funkcemi. Zvolení potřebných vývodů se provádí pomocí nastavení hodnot příslušného registru.

Piny 17 a 18 budou tvořit druhou I2C sběrnice, které je směřována na vnější komunikace. Je to komunikace s řídicím systémem robotu KAMbot.

Piny 8 a 10 v budoucnu budou naprogramované jako indikační vstupy, které budou detekovat přítomnost dvou signálů(signál tlačítka a signál z RTC).

Piny 11 až 14 jsou použité jako výstupy pro ovládání informačních LED diod.

Pin 9 bude fungovat jako vstup pro alert signál z balancovacího čipu a jako výstup pro ovládání toho čipu.

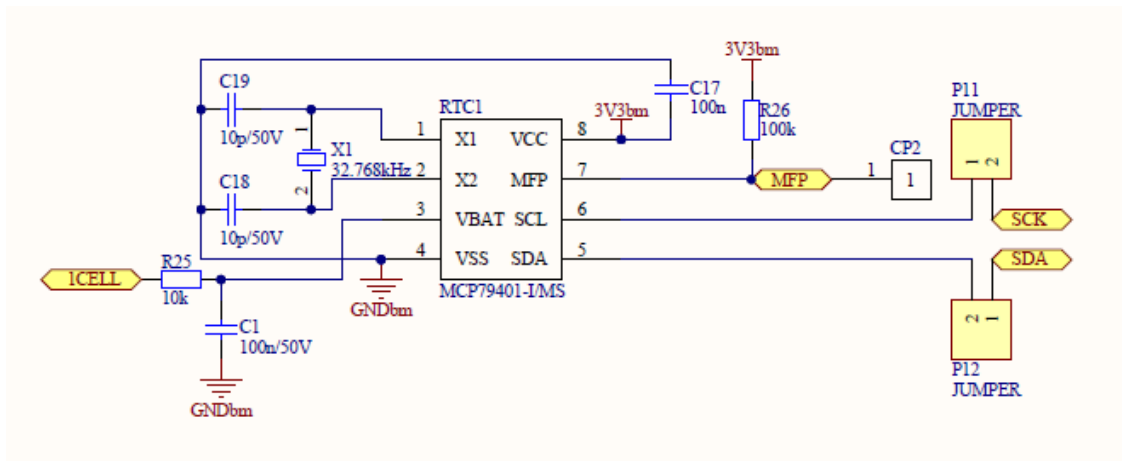
Z důvodu, že PCB je prototypem a je určená pro programování a ladění, celá deska je ucpána různými samostatnými kolíčky a jumpery. Je to z důvodu usnadnění odlaďování a kontroly důležitých signálů. Samostatné kolíčky ve schématu jsou poznamenané jako součástka CP(Check Pin). Jumpery jsou přidáné na vnitřní I2C sběrnice, která zaručuje komunikace MCU - RTC - Flash Paměti - Balancovacího čipu. Těmi jumpery se dá připojit jenom k jedné součástce, se kterou je potřeba komunikovat a pracovat jenom s ní. Ostatní čipy budou v klidu a odpojené od vnitřní I2C sběrnice, pokud se nepřidají příslušné propojky.

3.1.2 RTC

RTC je součástka, která se stará o přesnost a zachování běhu hodin při vypnutém napájení desky. Na PCB je použit čip od firmy Microchip s názvem MCP79401-I/MS. Hlavními výhodami této součástky jsou připojení k baterie, rozměr a možnost I2C komunikace.

Dříve bylo řečeno, že MCU má uvnitř implementované RTC a piny pro připojení hodinkového krystalu s frekvencí 32,768kHz. Bohužel ty piny jsou využité pro komunikace po I2C sběrnice. Dalším důvodem přidání externího RTC je ten, že microcontroller nebude schopen podporovat tuto funkci bez napájení. Obrázek 3.4 předvádí zapojení externího RTC čipu.

Na piny 1 a 2 se připojuje přesný hodinkový krystal s filtrací obou vývodů. Z důvodu použití jiného krystalu než těch, co bylo předloženo v datasheetu, hodnota



Obrázek 3.4: Zapojení RTC

kondenzátoru C19 je stejná s hodnotou kondenzátoru C18.

Piny 4 a 8 jsou napájecí a napájí se z PCB. Napájení čipu takže obsahuje filtraci pro vyhlazení a vyrovnání napětí.

Piny 5 a 6 jsou komunikační(I2C sběrnice).

Pin 3 stejně jako vývod 8 přivádí do čipu energie, ale bere to z prvního článku baterie. Proto při odpojení napětí na desce, RTC bude furt zapnuté a bude běžet díky tomu připojení rovnou na baterie. Tato napájecí cesta takže má filtraci. Napájení z tohoto pinu se začíná, jakmile se vypne napětí na desce, o tom přepnutí čip se stará sám. Rezistor R25 je přidán, aby zamezil proud tekoucí do čipu.

Pin 7 je výstupní pin, který má dvě funkce. První funkce je vysílání alarm signálu, který říká, že RTC jede v VBAT modu, a druhá funkce je vysílání clock signálu, který se dá detekovat a vědět, že RTC je funkční a je v provozu.

3.1.3 Flash Paměť

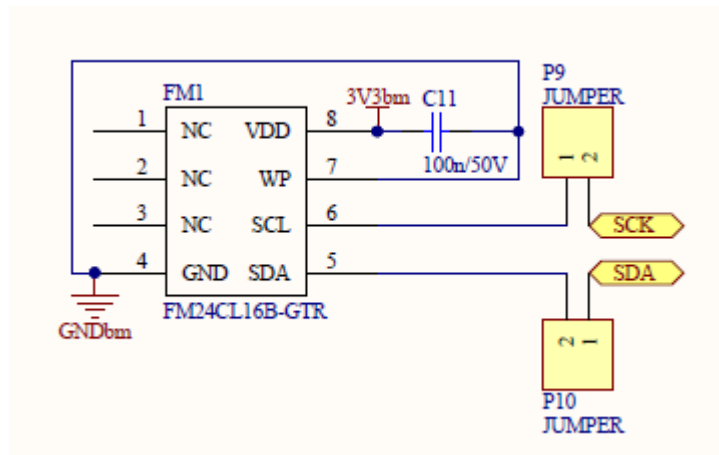
Externí Flash paměť s velikostí paměti 16kbit je implementovaná na desce, aby se tam ukládali užitečné data, což jsou například hodnoty článků těsně před odpojením a čas odpojení. Na obrázku 3.5 je vidět, jakým způsobem je zapojená paměť.

Piny 1 až 3 nemají žádnou funkci, proto jsou pojmenované jako NC.

Piny 4 a 8 se používají pro napájení čipu. Kondenzátor C11 je použit pro filtraci napájení.

Piny 5 a 6 jsou vyhrazené pro I2C komunikace s MCU.

Pin 7 je speciální pin, funkci kterého je ochrana před zápisem. V momentu když

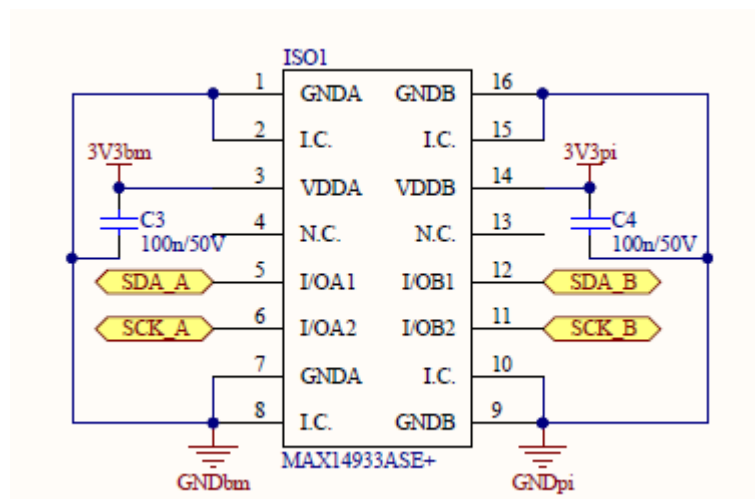


Obrázek 3.5: Zapojení Flash paměti

tento vývod je trvalé napájen, Flash paměť je zamknuta, což znamená, že se nedá provádět žádné změny a je povoleno jenom čtení. Když tento pin bude přiveden na zem, tak paměť čipu se dá i upravovat.

3.1.4 Digital Isolator

Ve finální verze balanceru, což není cílem práce, PCB musí komunikovat s řídicím systémem robotu. Z důvodu toho, že systém robotu je nejdůležitější část a s balancerem je propojena jenom po I2C sběrnice, tato komunikace musí být chráněna. To se dělá proto, že v případě odpálení balanceru, řídicí deska robotu zůstane nepoškozená. Ochranu zaručuje přidána oddělovací součástka. Je to digitální I2C izolátor od firmy Maximum Integrated s názvem MAX14933ASE+ pro ochranu I2C komunikace.



Obrázek 3.6: Zapojení I2C izolátoru

Piny 1, 3, 7, 9, 14 a 16 jsou použité pro napájení součástky s přidáním filtraci. Z důvodu toho, že je to oddělovací čip, levá a pravá strana součástky jsou napájené z různých zdrojů a mají rozdílné země.

Piny 4, 8, 10 a 15 jsou nestandardní piny se specifickými funkcemi. Vývody 4 a 13 jsou NC a nemají být zapojené. Piny 8 a 10 jsou IC a podle datasheetu mají být zapojené na příslušné jejich stranám země.

Piny 5 a 6, 11 a 12 jsou vstupní / výstupní vývody, které zaručují komunikaci mezi dvěma oddělenými oblastí.

3.1.5 Balancovací Integrovaný Obvod

Hlavní součástíou celé PCB desky je balancovací čip firmy Texas Instruments BQ7692006PW. Tento integrovaný obvod se používá pro pěti článkové baterie nebo pro baterky s menším počtem článků. Čip používá metodu pasivního balancování na základě napětí a podporuje interní a externí vyvažování. Tento integrovaný obvod je podobný BQ77PL900 čipu z teorie a vyvažuje způsobem, který je ukázán na obrázku 2.10. Tedy tento čip není schopen vyvažovat všechny články najednou.

Co se týče bezpečností tak, čip ochraňuje baterie na úrovni hardwaru a má zabudované v sobě 4 různé typy ochrany:

- proti přebíti
- proti podbití
- proti zkratu
- proti nadlimitnímu proudu při vybití

Tato součástka měří nejen napětí každého článku a protékající proud, takže se měří napětí celé baterie a teplota. Na obrázku 3.6 je zobrazeno zapojení balancovacího čipu a obvodu.

Piny 1 a 2 ovládají dva velké N-MOSFET tranzistory, které jsou připravené pro odpojování zátěží při porušení aspoň jednoho typu ochrany.

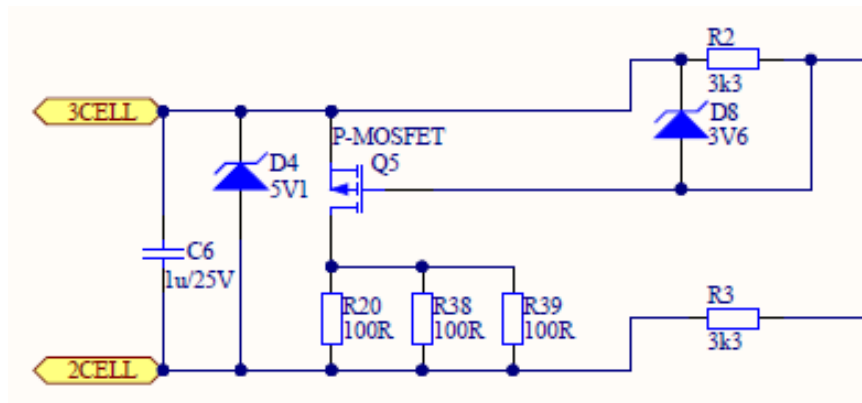
Piny 3 a 10 řeší napájení součástky s přídatnou filtrací.

Pin 11 se nepoužívá a je NC, takže se nikam nezapojuje.

Piny 4 a 5 jsou stanovený pro komunikace. Dány integrovaný obvod taky má jumpery na vnitřní I2C sběrnice a dá se jich použít nejen pro ladění. Takže je možnost připojení tohoto čipu na rovnou na externí I2C sběrnice.

Piny 8 a 9 řeší LDO regulátor. Vývod 9 je přívodem s filtrací a bere napětí rovnou z baterii. Pin 8 je výstupem ze součástky a má hodnotu napětí 3,3V. Tento vývod se používá pro napájení řídicího MCU nebo LED diod. V případě balanceru microcontroller, flash paměť a RTC se napájí z tohoto vývodu.

jené několik stejných rezistorů, které přes sebe začnou propouštět část nabíjecího proudu. Nabíjení se nezastaví, jenom se trochu přibrzdí. Hodnoty balancovacích rezistorů se většinou pohybují v rozmezí 50 - 80Ω. Na prvním prototypu jsou použité 100Ω rezistory z důvodu toho, že v moment pájení byli v dílně a nebyla potřeba testovat balancování. Jestli by místo rezistoru se použilo řešení s výkonovými tranzistory, které převedou praktický veškerou nabíjecí energii v energii tepelnou. Byla by potřeba řešit chlazení. Balancovací odpory v tomto případě jenom sniží nabíjecí proud a převedou v teplo jen malou část energii. Rezistory se moc nezahřívají, protože v teplo se mění jenom malá částka energie. Vyzářené teplo stihne se ztratit v okolním prostředí a není potřeba mít žádný radiátor na chlazení. Balancovací proud tohoto zapojení má maximální hodnotu cca 150mA. Hodnoty vyvažovacích proudu se obvyklé pohybují v rozmezí 100 - 300mA. Jestli nabíjecí proud bude o hodně vyšší než balancovací proud, tak při velkém rozbalancování článků balancer nestihnou vyrovnat napětí na člancích do doby, kdy nejlépe nabity články dosáhne hodnoty 4,2V.[5]



Obrázek 3.8: Zapojení balancovacího obvodu

Při návrhu zapojení z obrázku se uvažovalo o tom, kolik rezistorů je potřeba mít při maximálním balancovacím proudu 150mA. Dalším bodem bylo to, že ty rezistory mají být stejné a mít footprints 0603 nebo 1206. Řešilo se čtyři možnosti:

- použít 1 rezistor
- použít 2 rezistory
- použít 3 rezistory
- použít 4 rezistory

Před tím jak vybírat rezistory, oni se vyhledali, brali se od 50Ω do 80Ω , a vypočítali se výkony těch rezistoru podle vzorce:

$$P = I^2 * R$$

$$P = (50 * 10^{-3} A)^2 * 50\Omega = 1,25W$$

Po těch výpočtech pak vzniklá tabulka výkonu rezistorů. Takže je samozřejmostí,

že v paralelním zapojení stejných rezistorů proud se rozdělí na stejné části.

Počet	Proud	Odpor	Výkon	Pouzdro	Výkon Výrobce
1	150mA	50Ω	1,125W	-	-
2	75mA	51Ω	0,2869W	1206	0,5W
2	75mA	75Ω	0,4219W	1206	0,5W
3	50mA	51Ω	0,1275W	1206	0,25W/0,5W
3	50mA	75Ω	0,1875W	1206	0,25W/0,5W
4	37,5mA	51Ω	0,0717W	0603/1206	0,1W/0,125W/0,25W/0,5W
4	37,5mA	54,9Ω	0,0772W	0603/1206	0,1W/0,125W/0,25W
4	37,5mA	56Ω	0,0788W	0603/1206	0,1W/0,125W/0,25W
4	37,5mA	60,4Ω	0,0849W	0603/1206	0,1W/0,125W/0,25W
4	37,5mA	61,9Ω	0,0871W	0603/1206	0,1W/0,125W/0,25W
4	37,5mA	62Ω	0,0872W	0603/1206	0,1W/0,125W/0,25W
4	37,5mA	64,9Ω	0,0913W	0603/1206	0,1W/0,125W/0,25W
4	37,5mA	68Ω	0,0956W	0603/1206	0,1W/0,125W/0,25W
4	37,5mA	75Ω	0,1055W	1206	0,25W/0,5W

Tabulka 3.1: Výkony rezistorů

Z dopočítaných hodnot výkonů je vidět, že použití jednoho rezistoru je strašně náročná věc, protože pouzdro takového rezistoru bude 2512, což je obrovská součástka a bude silně hřát, toto řešení není možné použít. Kromě zvolení součástky se správným výkonem vzniklá potřeba vypočítat maximální napětí, které může být přivedeno. Ten výpočet se prováděl podle vzorce:

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

$$U = \frac{0,5W}{75 * 10^{-3}A} = 6,667V$$

Počet	Proud	Výkon	Napětí
2	75mA	0,5W	6,667V
3	50mA	0,25W	5V
3	50mA	0,5W	10V
4	37,5mA	0,125W	3,333V
4	37,5mA	0,25W	6,667V
4	37,5mA	0,5W	13,333V

Tabulka 3.2: Napětí příslušné proudům rezistorů

Z tabulky 3.2 je vidět, že variantu s čtyřmi rezistory, které jsou v pouzdru 0603 a výkonem 0,125W, nejde použít, protože maximální napětí při tomto proudu a výkonu je menší než 4,2V a je to napětí plně nabitého článku.

Optimální variantou by bylo použít tři rezistory s footprintem 1206 a odporem 51Ω . Ale při pájení, jak už dřív bylo řečeno, se použili rezistory 100Ω . Ve výpočetních tabulkách součástka s takovou hodnotou odporu není, proto je potřeba provést předcházející výpočty pro tento rezistor.

$$P = (50 * 10^{-3} A)^2 * 100\Omega = 0,25W$$
$$U = \frac{0,25W}{50 * 10^{-3} A} = 5V$$

Po dokončení výpočtu je vidět, že hodnota výkonu je $0,25W$, tedy bylo by ideální použít rezistor s výkonem $0,5W$. Bohužel není známo, jakou hodnotu výkonu mají použité rezistory. Díky internetovým elektro obchodům se dalo dozvědět, že rezistory v pouzdru 1206 mají minimální hodnotu výkonu stanovenou výrobcem $0,25W$. V této situaci se dá vycházet z toho, že použité rezistory mají výkony $0,25W$, což není dobré. V nejhroším případě ty rezistory se znehodnotí během provozu balanceru.

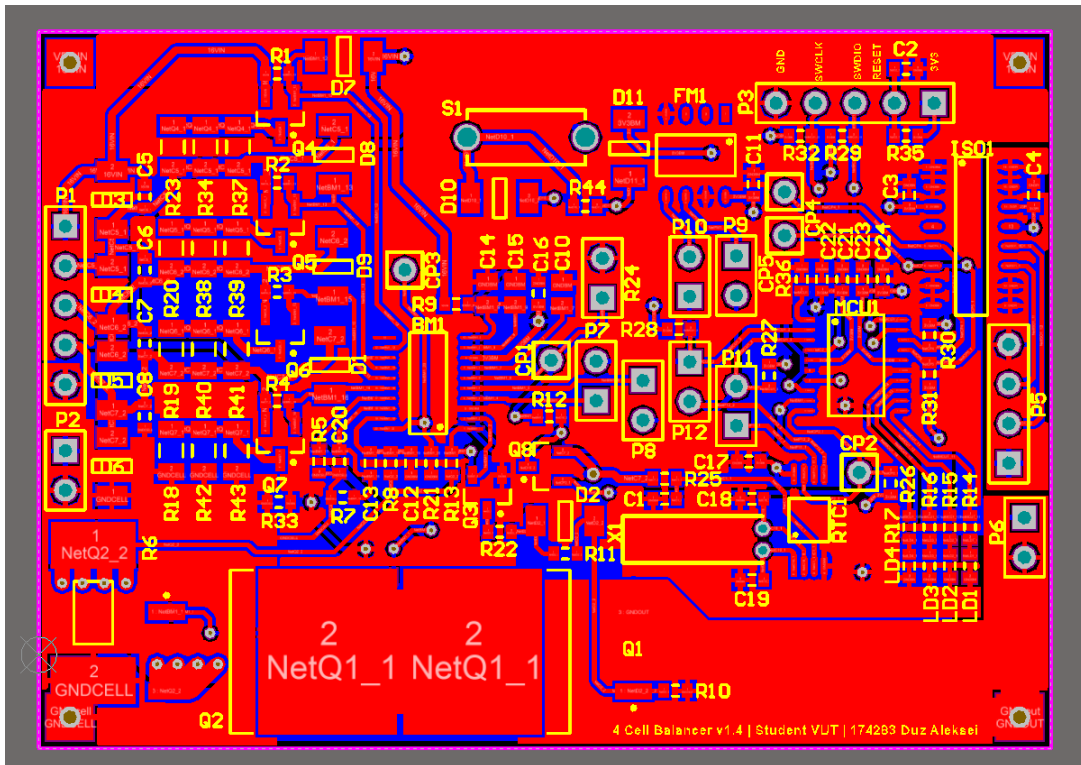
3.2 Připomínky k PCB desce

Balancer byl navržen a rozveden v PCB programu AltiumDesigner16. Při návrhu a vytvoření schémat se používali referenční zapojení čipů s drobnou úpravou některých hodnot pasivních a polovodičových součástek.

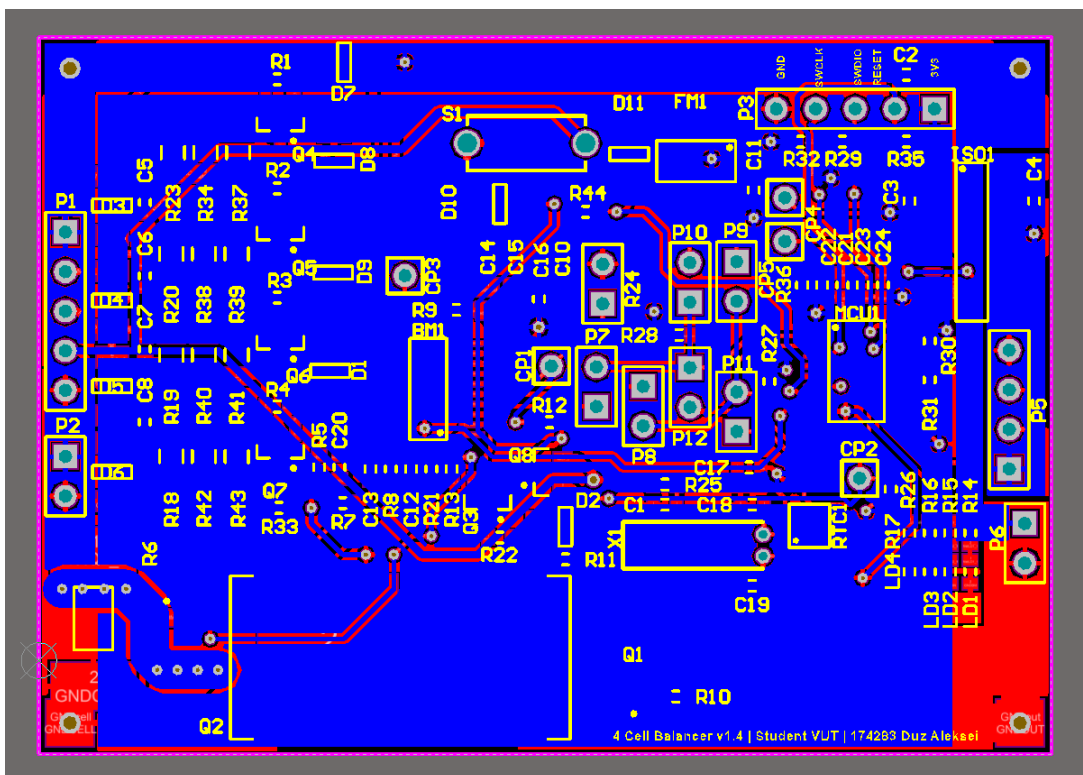
Během programování, pájení a ladění desky byli odhaleny další chyby a vyměněné součástky:

- pouzdra řídicích N-MOSFET tranzistorů jsou moc velké; z jedné strany je to výhoda, sílové tranzistory mají velkou chladičskou plochu, ale z druhé strany by se dalo PCB zmenšit o několik mm
- chyba zapojení programovacího konektoru P3; programovací signál SWCLK musí mít místo pull-up rezistoru pull-down (odpor R32)
- tranzistor Q8 má chybné zapojení vývodů drain a source; mají být prohozené
- všechny pull-up a pull-down rezistory mají hodnotu odporu $12k\Omega$ místo navržených $10k\Omega$; důvod je nedostatek $10k\Omega$ rezistorů při pájení
- špatnou stranou napájeny LED diody 1 až 3 není dopájen konektor pro vnější komunikaci P5.

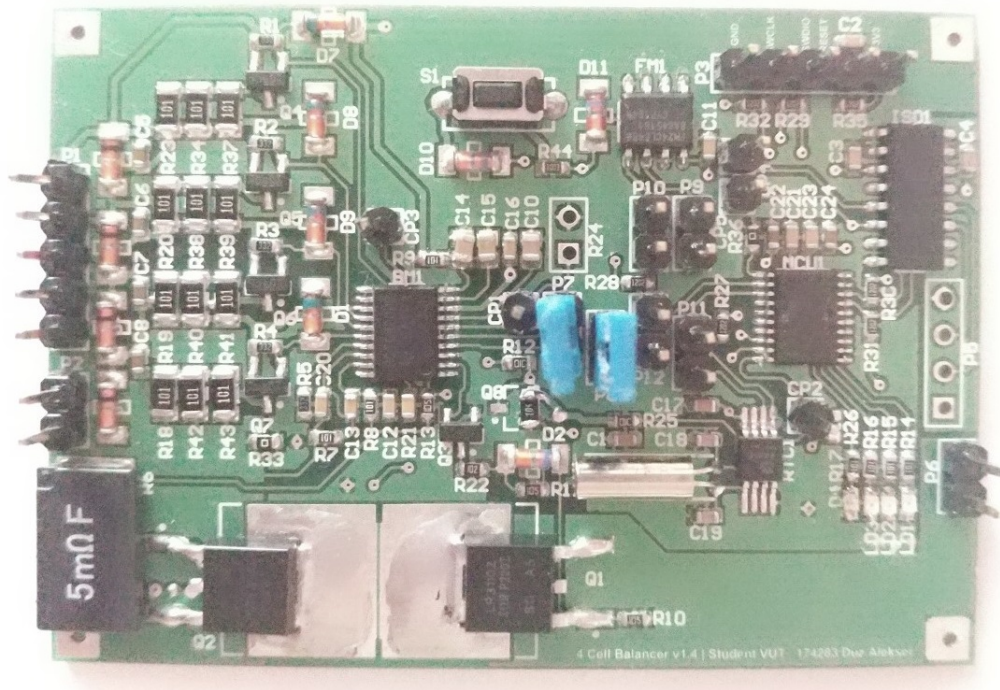
Při programování bylo nutné balancer napájet a to takovým způsobem, aby se dalo otestovat čip BQ7692006PW. Jestli byla použita Li-Ion baterie, možno už by byla zničená. Proto se vytvořila virtuální baterka, která se furt napájela ze zdroje napětí. Konektorem pro připojování článků na desce byla P1.



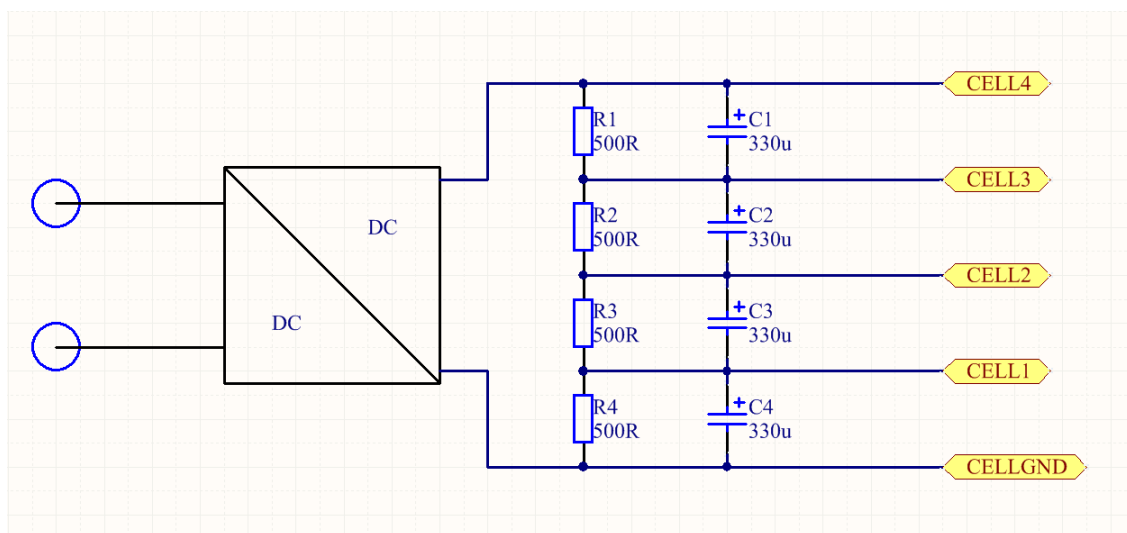
Obrázek 3.9: Top vrstva balanceru



Obrázek 3.10: Top vrstva balanceru



Obrázek 3.11: Zapájený balancer



Obrázek 3.12: Schéma virtuální baterie se zdrojem

Kapitola 4

Software

4.1 Použitý software a pomůcky

Pro programování microcontrollerů se používá speciální programovací prostředí a také speciální přístroj, který se nazývá programátor.

V dnešní době existuje spousta programovacích prostředí, která podporují ten nebo jiný čip. Pro člověka, který se s tím setkává poprvé, je docela velký problém si vybrat jedno z desítek různých vývojových prostředí, protože každé z nich má své výhody a nevýhody. Pro začátečníka je lepší si zvolit prostředí s IDE, protože takový software má shrnuté nástroje programovacího jazyka do ucelené podoby, která urychluje pochopení jazyka a zvětšuje výkon vývojáře. IDE prostředí obsahuje editor zdrojových kódů, kompilátory a občas debugery, je to nástroj, který umožňuje krokovat a ladit napsaný program. Je potřeba ještě říct, že vývojové prostředí většinou jsou zaměřené na určitý programovací jazyk.

Některé výrobci microcontrollerů pro své zákazníky vyvíjí speciální pomocné programy, které umožňují urychlit napsání kódu, sledovat proměnné za běhu programu a nahrávat kód do čipu. Program pro urychlení kódování od firmy STMicroelectronics se nazývá STM32Cube. To je grafické uživatelské rozhraní, ve kterém si zvolíte používané MCU, naklikáte potřebné funkce (komunikace, GPIO atd.) a program Vám vygeneruje kód s komentáři, do kterého se budou doplňovat ovládací funkce. Je to vytvořeno pro lidi, které s programováním MCU se setkávají poprvé. Další software umožňuje kontrolovat změny proměnných za běhu programu. Takové programy se vytváří, protože ne všechny vývojové prostředí mají funkci debug. STMicroelectronics nazývá takový program STMStudio. Poslední pomocný software, který je nabízen touto firmou, je program pro nahrávání napsaného kódu do controlleru a on má název ST-Link Utility.

Co se týče programátorů, tak jsou to zařízení (není jich hodně oproti prostředím), která vytvořené výrobci MCU pro svou produkci a které mají za úkol propojit počítač a microcontroller, aby se dalo nahrát napsaný kód do čipu a mít zpětnou vazbu při ladění.

4.1.1 IDE vývojové prostředí

Pro programování čipu STM32F042F6P6 byli vyhledané další vývojová IDE prostředí:

- SW4STM32(AC6)
- CoIDE(CooCox)
- MDK-ARM-STM32(Keil)
- CosmicIDE(Cosmic)
- EmBitz 1.11

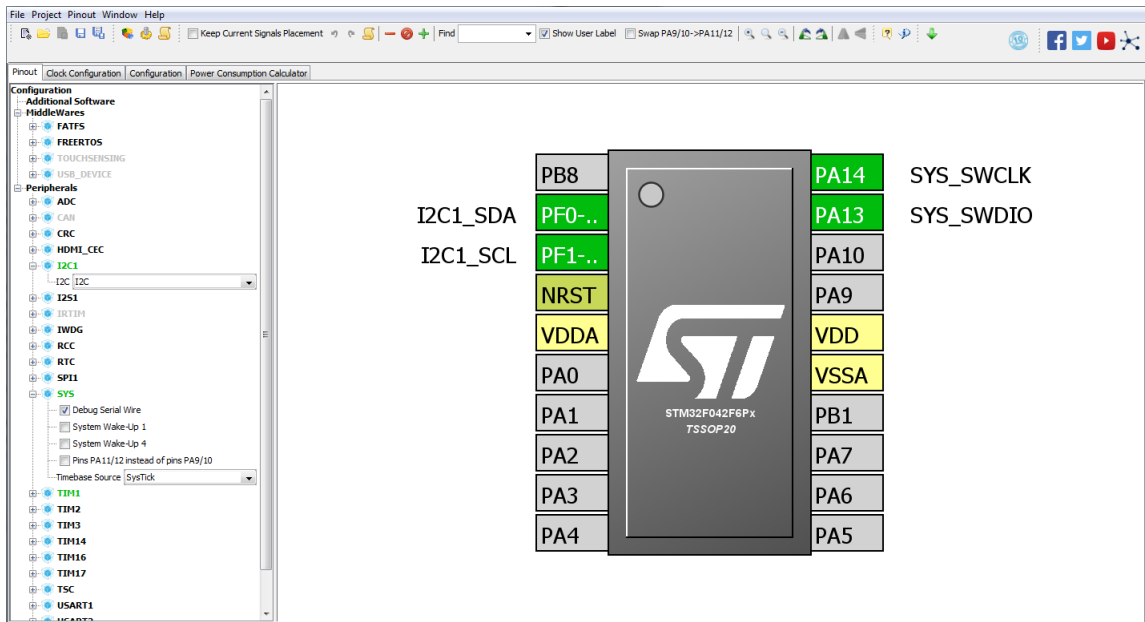
V této bakalářské práci pro programování MCU se používal software SW4STM32(Eclipse), protože byl dostatečně intuitivní při ovládání, měl vzorce programů od STMicroelectronics a vytvářel soubor s příponou .bin, který se pak nahrával do microcontrolleru. Bohužel toto vývojové prostředí neobsahovalo debug nástroj, takže prakticky se nedalo ladit program. Programovací prostředí EmBitz 1.11 umožňovalo programovat STM čipy s laděním. Program byl zamítnut, protože byli drobné problémy s instalací, nebyl čas na porozumění programu, protože software byl nalezen na konci práce, a MCU už v tu dobu mělo nahrány program odpovídající zadání práce. V budoucnu se bude používat EmBitz, když se odstraní problémy instalací.

4.1.2 STM32Cube

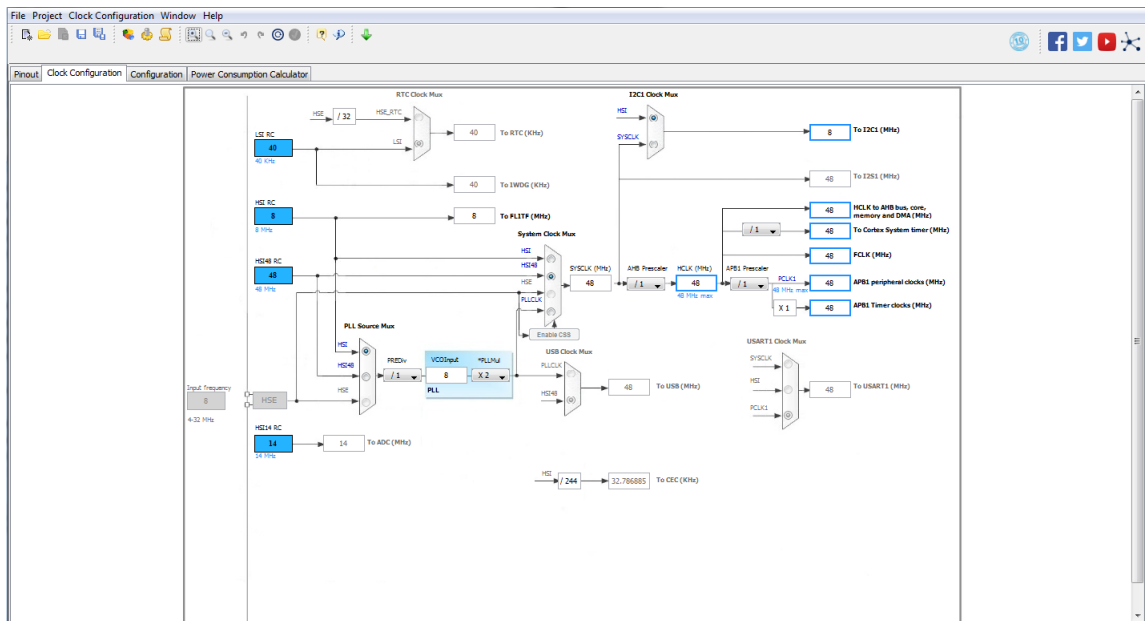
STM32Cube, jak dříve bylo řečeno, je program generující kód podle zvolených funkcí. Program je dostatečně jednoduchý a pro získání kódu je potřeba provést následující kroky:

- zvolit si správně rodinu, řadu a čip MCU
- kliknout nahoře na Project->Settings
- zadat název projektu a nastavit místo v počítači, kam se uchová vytvořený kód
- nastavit IDE, které budete používat pro programování
- nastavit v záložce Code Generate potřebné parametry generování a zavřít Settings stiskem OK
- nastavit si potřebné piny a jejich funkce(program pomůže různými radami a upozorněními)
- zkontrolovat clock nastavení(jaký oscilátor se bude používat a jeho frekvence)
- kliknout nahoře na Project->Generate Code

V nastavené složce se vygeneroval projekt pro nastavené IDE a může se začít programování.



Obrázek 4.1: Ukázka grafického prostředí STM32Cube



Obrázek 4.2: Nastavení clock konfigurace v STM32Cube

4.1.3 ST-Link Utility a programátor ST-Link V2

STM32 ST-Link Utility je program, který nahrává vygenerovaný vývojevým prostředím soubor s příponou .bin do MCU a spouští běh programu. Soubor s .bin je dokument, ve kterém je výsledek převodu napsaného kódu do bináru.

Sama o sobě tato utilita není schopná nahrát kód, k tomu potřebuje programátora.

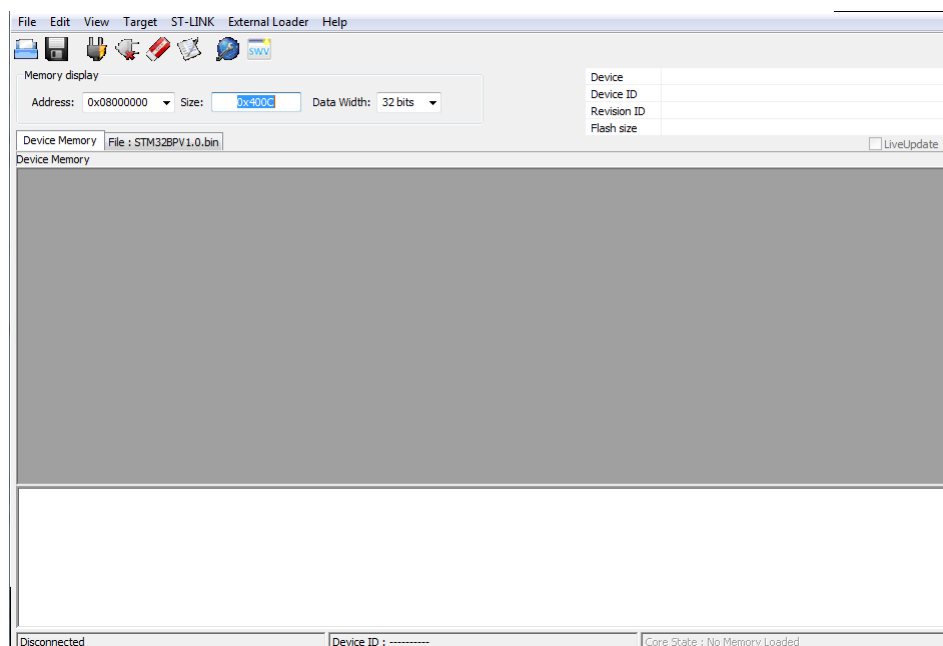
Na to byl použit ST-Link V2. Programátor si přečte binární soubor, přeloží ho na jazyk MCU a nahraje ten kód do správného místa v paměti microcontrolleru.

Programátor k balanceru se připojí k programovacímu konektoru P3 z konektoru CN3 pomocí propojovacích drátů.

Pin PCB	Název Pinu PCB	Pin ST-Link	Název Pinu ST-Link
1	3V3	1	VAPP
2	RESET	15	NRST
3	SWDIO	7	SWDIO
4	SWCLK	9	SWCLK
5	GND	16	GND

Tabulka 4.1: Připojení programátoru k balanceru

Pin VAPP programátoru nenapájí desku 3,3V a jenom detekuje přítomnost napájecího napětí.



Obrázek 4.3: Uživatelské rozhraní programu ST-Link Utility

Postup nahrávání kódu do MCU je následující:

- připojit se k MCU kliknutím na Target->Connect
- připravit se k nahrávání kódu kliknutím na Target->Program Verify
- vybrat správný binární soubor
- zvolit si možnost Verify after programming
- kliknout Start

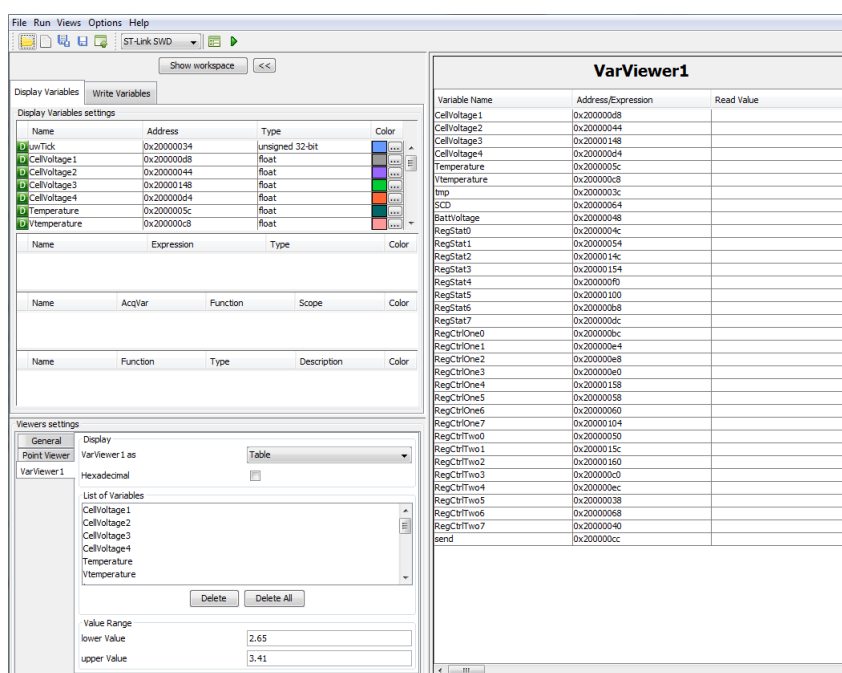
- po dokončení nahrávání zavřít program

Po nahrání programu microcontroller se restartuje automaticky.

4.1.4 STMStudio

Z důvodu toho, že se používal Eclipse, který nemá debug nástroj, byla potřeba najít nějakou náhradu na čtení proměnných z MCU za běhu nahraného programu. Pro tyto potřeby STMicroelectronics má vyvinuty software, který se nazývá STMStudio.

Tento program ke své činnosti potřebuje soubor s příponou .elf, který se vytváří spolu s binárním souborem, a s toho souboru získává názvy proměnných a jejich adresy.



Obrázek 4.4: Uživatelské rozhraní programu STMStudio

Postup přípravy programu k sledování změn proměnných je následující:

- nahrát soubor .elf kliknutím na File->Import variables
- vybrat všechny proměnné kliknutím na Select all, a pak Import
- zavřít okno nahrávání
- v okně Display Variables settings si vybrat proměnné
- pravým tlačítkem myši kliknout na vybrané řádky a zvolit Send To VarViewer1

Sledování změn proměnných v reálném čase se spouští stiskem tlačítka Start recording session. Je potřeba si pamatovat, že při zapnutí sledování programátor musí být volný(žádný program s ním nepracuje).

4.2 Programování

Cílem bakalářské práce bylo napsat kód, který provede nastavení registrů v balancovacím čipu BQ7692006PW po I2C sběrnice. Před tím, jak se začne rozebírat napsaný kód, musí být rozebráno, jaké registry mají být nastavené v integrovaném vyvažovacím obvodu a jak má být nakonfigurován microcontroller.

4.2.1 MCU

Při programování kódu vývojář musí rozumět tomu, jak funguje MCU, co se má nastavit, aby fungovali vývody, jaké mají parametry, a jaké funkce mohou být naprogramované na určitých pinech. Proto existuje referenční manuál s popisem funkcí celého čipu a s různými vzorci pro napsání programu.

Výrobce většinou se snaží ulehčit programování MCU, proto vyvíjí různé knihovny, díky kterým se dá nastavit příslušné registry pomocí speciálních funkcí a struktur.

RCC

RCC je obecný název pro skupinu registrů, které ovládají frekvenci hodinového signálu, jeho původ a různé možnosti restartování microcontrolleru. V dané skupině registru důležitou roli hrají clock registry.

Je potřeba vysvětlit několik zásadních věcí. Jakákoliv komunikace není možná bez přítomnosti signálu clock, který pomůže jasně interpretovat každý bit. Jeden bit se rovná jednomu ticku hodinového signálu a frekvence ticknutí stanoví rychlost komunikace.

MCU i bez uživatelského kódu má již od začátku nahraný nějaký program, který se nazývá firmware. Tento program je uložen ve vnitřní paměti microcontrolleru a po spouštění se začíná vykonávat. Přečte nastavení registrů, spustí procesor a přesměruje se do paměti, kam se nahrává uživatelský kód.

Aby se přečetli instrukce z paměti je potřeba mít vnitřní komunikace, která pro svou činnost potřebuje hodinový signál. MCU má vždy uvnitř implementovaný RC oscilátor, který i vytváří potřebný hodinový signál s určitou frekvencí, která už od začátku je dostupná procesoru. Když procesor běží na tomto signálu, on se nazývá systémovým hodinovým signálem.

MCU STM32F042F6P6 má uvnitř implementované dva RC oscilátory, jeden má hodnotu frekvence 8MHz a druhý má 48MHz. Tyto dva oscilátory díky zabudování do čipu se nazývají HSI RC oscilátory, respektive HSI a HSI48 oscilátory. Při spouštění microcontrolleru se nastavuje HSI oscilátor jako zdroj hodinového signálu pro systémový clock.

Nastavení systémového clock se dá změnit při programování, existuje čtyři možnosti zdrojů systémového hodinového signálu:

- HSI
- HSI48
- HSE
- PLL

HSE oscilátor může být nastaveno jenom v případě jestli je použit vnější oscilátor a je připojen na speciálně stanovené na to piny MCU. PLL je násobič nebo dělič pro vstupní do něho frekvenci a na výstupu bude mít větší nebo menší hodnotu kmitočtu. Dělení nebo násobení se provádí pomocí celý čísel.

Je nutno si ujasnit, že systém microcontrolleru si zvolí jeden z čtyř zdrojů hodinového signálu. Co se týče vnějších periférií a vysílání hodinového signálu z čipu, tak zdrojů signálu clock je podstatně více.

Pro optimalizace energetické spotřeby na začátku MCU má vypnuté převádění clock signálu na vnější periférii, což jsou piny microcontrolleru. Proto aby fungovali vnější vývody a vykonávali naprogramované funkce, je potřeba v RCC registrech povolit přivádění hodinového signálu na používané piny respektive porty.

GPIO

Z hlediska MCU GPIO je skupina registrů, která ovládá funkce vnějších vývodů, jejich nastavení a provádí kontrolu.

Z hlediska PCB desky GPIO se nazývají vnější vývody microcontrolleru. Ty piny se používají pro komunikace čipu s vnějším prostředím a umožňuje jeho ovlivňování. To se netýká pinů, které mají nějaké specifické funkce například napájení čipu nebo externí reset signál.

Je nutno si ujasnit, že piny MCU jsou programovatelné a je potřeba na začátku si zvolit, na co se bude používat vybraný vývod. Buď to přijímání signálu, vysílání signálu nebo nějaká speciální či alternativní funkce typu komunikace po sběrnice. Takže je dobře vědět, že výrobce taky o tom přemýšlí a některé piny už mají naprogramované určité alternativní funkce například programovací piny nebo piny pro I2C komunikace.

Při programování microcontrolleru už na začátku je jasné, co má dělat každý vývod čipu. To dáno zapojením PCB desky a zbývá jen naprogramovat nebo využít jednu z nachystaných alternativních funkcí tohoto pinu.

I2C sběrnice

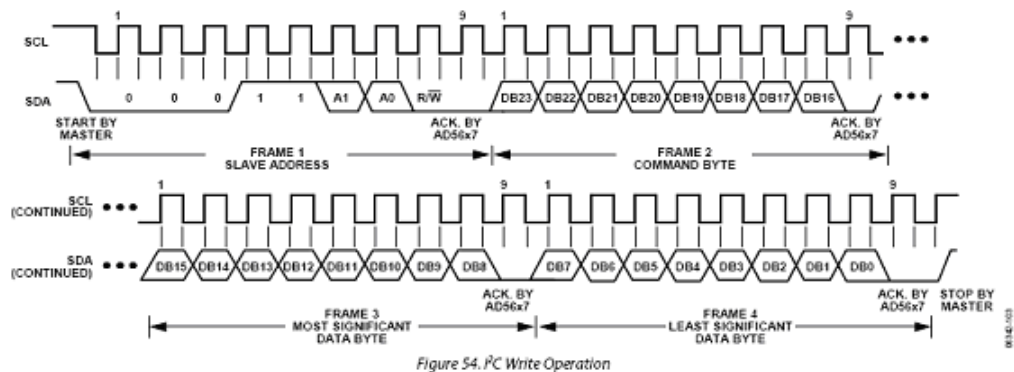
I2C sběrnice je sériová sběrnice, která podporuje multi-master technologii (připojení několika mastrů na jednu sběrnice) a používá se pro připojení nízkofrekvenčních periférií. Sběrnice se skládá ze dvou vodičů SCL(clock z mastru) a SDA(data). V dnešní době I2C komunikace má tři rychlostní módy, jsou to komunikace s kmitočtem hodinového signálu 100kHz, 400kHz a 1MHz. Bohužel rychlost sběrnice je

stanovena nejpomalejší součástkou na ní. Pro navržený balancer rychlost I2C sběrnice je 100kHz, tato hodnota je dána balancovacím čipem BQ7692006PW.

Existuje několik módů provozování I2C sběrnice:

- slave odesílatel
- slave příjemce
- master odesílatel
- master příjemce

Na začátku komunikace čip běží ve slave modu. Interface automaticky se přepne ze slave modu do master při generování START bitu a zpátky při generování STOP bitu nebo ztrátě arbitráže. Komunikace funguje následujícím způsobem.



Obrázek 4.5: Komunikace po I2C sběrnice

1. Nejdřív se zkontroluje či sběrnice je v klidovém stavu. To se dá poznat ze signálu SCL a SDA a to tak, že v klidovém stavu mají high díky pull-up rezistorům.
2. Prvním se vysílá START bit, které zahájí komunikaci. Po detekování Start bitu se aktivuje hodinový signál sběrnice, který je generován mastrem.
3. Po START bitu následuje 7-mi bitová adresa příjemce zprávy a osmy bit řekne o tom, co bude probíhat, čtení nebo zápis. Takže je potřeba si zvolit mód adresování, je jich dva, 7-mi bitové adresování a 10-ti bitové adresování. Adresa vždy se odesílá v Master modu.
4. Pak následují bity, které nastaví pointer v paměti příjemce, kam se budou zapisovat data.
5. Odesílání připravených dat.
6. Čekání na acknowledge od příjemce o tom, že data byli přijatá. Jestli přijímací čip vyžaduje kontrolní součet po dokončení odesílání dat, je potřeba ho poslat a pak čekat na acknowledge. Jestli kontrolní součet nebude odeslán, příjemce přijatá data neuloží do paměti. Nutnost kontrolního součtu záleží na čipu příjemce.

Integrovaný obvod BQ7692006PW při komunikaci po I2C sběrnice nevyžaduje kont-

rovní součet po přijetí dat z MCU. Výše popsaný způsob komunikace se používal při práci s tímto čipem. Takže je nutno říct, že při komunikaci mezi STM32F042F6P6 a BQ7692006PW MCU odesílalo a přijímalo data v master modu.

4.2.2 Balancovací čip

Registry

Balancovací čip je už naprogramován výrobcem a vývojář musí jenom provést nastavení příslušných registrů, jinak řečeno, provést konfiguraci.

Aby balancovací integrovaný obvod byl funkční, mají se nastavit následující parametry:

- musí se povolit měření napětí a napětí(ADC)
- musí se povolit měření proudu(CC)
- musí se nastavit horní a dolní limity napětí pro články
- musí se nastavit hodnoty proudu při zkratu a maximální hodnota proudu při vybití
- musí se povolit ochrana pomocí charge FET a discharge FET

Na začátku všechny registry balancovacího integrovaného obvodu jsou v nule, aby se nastavilo potřebné chování čipu, registry se mají přepsat na kombinaci nul a jedniček. Z důvodu toho, že tato součástka má specifickou funkci, ona má o hodně menší počet registru než MCU. Takže je snadnější se hrabat v těch registrech a nastavovat jich na přímo. Každý registr toho čipu je popsán v datasheetu a má popsané i nastavitelné hodnoty registrů.

Registr SYS_STAT je status registr, ze kterého se dá vyčíst, která chyba nastala. Buď to podpětí nebo něco jiného. Je potřeba na začátku programu ho vynulovat, nemá automatické nulování.

Registry CELLBALx se používají pro balancování baterie, zapnutím a vypnutím příslušných kanálu se řídí vyvažování.

Registry SYS_CTRLx jsou registry, které zapínají různá měření a ochrany. Tyto registry bude potřeba v programu nastavit. V SYS_CTRL1 se nastaví ADC_EN a TEMP_SEL, tím se zapne měření napětí a teploty, měření teploty se přepne do měření externím termistorem a zapne se ochrana proti přebíti. V SYS_CTRL2 se nastaví CC_EN, DSG_ON a CHG_ON, tím se zapne měření proudu a automatické ovládání rozpojovacích tranzistorů N-MOSFETu.

Registry PROTECTx jsou registry pro nastavení hodnot, při dosažení kterých se má rozpojit napěťový okruh, což je zapnutí ochrany baterii. V PROTECT1 se nastaví RSNS, SCD_Dx a SCD_Tx bity, tím se nastaví rozsah používaných hodnot, doba rozpojení a určí se hodnota zkratového proudu. Dále v PROTECT2 se nastaví OCD_Dx a OCD_Tx, tím se určí hodnota maximálního vybíjecího proudu a doba rozpojení. A v PROTECT3 se nastaví UV_Dx a OV_Dx, tím se určí doby rozpojení

obvodu při přebíti a podbití.

Registr `OV_TRIP` svým nastavením provede upřesnění horní hranici nabíjení, což znamená, že tím se nastaví limitní hodnota přebíti(4,2V). Je potřeba nastavit, aby čip fungoval správně.

Registr `UV_TRIP` nastaví hodnotu dolní hranici vybíjení, což by měla být hodnota 3,7V podle normy, ale tento čip to neumožňuje. Maximální možná hodnota, kterou se dá nastavit je 3,1V. V programu byla nastavena hodnota dolní hranice cca 3V, ale z důvodu, že při programování a ladění k dispozici byl jenom zdroj napětí na 12V, se ponechala původní hodnota dolní hranice.

Registr `CC_CFG` je potřeba nastavit pro optimalizace funkčností čipu.

Registry `VCx_HI` a `VCx_LO` se používají pro měření napětí na článcích. V napsaném programu se tyto registry používají pro ladění, kontrolu běhu čipu a otestování správností převodu získané hodnoty na napětí.

Registry `BAT_HI` a `BAT_LO` se používají pro měření napětí celé baterie.

Registry `TSx_HI` a `TSx_LO` jsou stanovené pro získání informací o teplotě článků(baterie).

Registry `CC_HI` a `CC_LO` se používají pro měření proudu.

Registry `ADCGAINx` a `ADCOFFSET` jsou stanovené pro získání hodnot `GAIN` a `OFFSET`, které se používají v převodech získaných hodnot na napětí. V programu tyto hodnoty byli použité jako statické, proto výpočet není přesný.

Výpočty ochran

Cílem bakalářské práce je nastavit hodnoty ochran v balancovacím čipu.

Před tím jak se začnou provádět výpočty je potřeba stanovit hodnoty parametrů `GAIN` a `OFFSET`. Jestli brát to poctivě, tyto hodnoty se mají získávat z příslušných registru, ale pro začátek se budou používat statické parametry pro pochopení funkčností čipu. Pak ve finální verze se to doladí správným způsobem. `GAIN` se bude rovnat 365 uV/LSB a `OFFSET` bude 1mV.

1) Hodnota horního napětí - ochrana proti přebíti(max. 4,2V)

$$OxAC = 10101100_{(2)}$$

Při převodu na napětí tato binární hodnota se doplňuje dalšími bity ve tvaru 0010-XXXXXXXX-1000.

$$ADC = 0010101011001000_{(2)} = 10952$$

$$V_{CELL} = ADC * GAIN + OFFSET$$

$$V_{CELL} = 10952 * 365 * 10^{(-6)} + 0,001 = 3.998V$$

$$ADC = \frac{4,2 - 0,001}{365 * 10^{(-6)}} = 11504$$

$$ADC = 11504 = 0010110011110000_{(2)}$$

Binární výsledek neodpovídá vzorce, proto se udělá kalibrace:

$$ADC1 = 11512 = 0010110011111000_{(2)}$$

$$ADC2 = 11496 = 0010110011101000_{(2)}$$

Hodnota ADC2 je v desítkové soustavě blíž ADC a hodnota napětí bude trochu menší než 4,2V. Z teorii se dozvědělo, že radši mít článek baterie o několik procent nedobitým než furt ho přebíjet. Proto se zvolí hodnota ADC2, odstraní se doplňující bity, vypočítá se hexadecimální hodnota a nastaví se do příslušného registru.

$$11001110_{(2)} = 0xCE$$

2) Hodnota dolního napětí - ochrana proti podbití(min. 3V)

$$0x97 = 10010111_{(2)}$$

Při převodu na napětí tato binární hodnota se doplňuje dalšími bity ve tvaru 0001-XXXXXXXX-0000.

$$ADC = 0001111111110000_{(2)} = 6512$$

$$V_{CELL} = 6512 * 365 * 10^{(-6)} + 0,001 = 2,378V$$

Je to strašné malá hodnota napětí a při ní články se ničí. Nejvyšší hodnota dolní hranice je 3,1V, proto 0xFF je nejvyšší hodnota, kterou se dá nastavit.

$$ADC = 0001100101110000_{(2)} = 8176$$

$$V_{CELL} = 8176 * 365 * 10^{(-6)} + 0,001 = 2,985V$$

Je vidět, že i s použitím hodnoty 0xFF napětí je nižší než 3V, eliminuje se to použitím většího GAINu.

3) Zkratový proud

Zkratový proud se počítá z odporu proudově-citlivého rezistoru a nastaveného zkratového napětí. Při výpočtu se používá Ohmův zákon.

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

$$I = \frac{22 * 10^{(-3)}}{5 * 10^{(-3)}} = 4,4A$$

Je to teoretická hodnota, která musí být otestovaná.

4) Maximální vybíjecí proud

Maximální vybíjecí proud se taky vypočítává z odporu proudově-citlivého rezistoru a nastaveného zkratového napětí. Při výpočtu se používá Ohmův zákon.

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

$$I = \frac{8 * 10^{(-3)}}{5 * 10^{(-3)}} = 1,6A$$

Je to teoretická hodnota, která taky musí být otestovaná.

4.3 Rozbor programu

4.3.1 Inicializace

Skelet kódu byl vygenerován programem STM32Cube, tím programem byli vygenerované funkce zapnutí hodinového signálu na určených portech, inicializovaná komunikace po I2C sběrnice a nastaven systémový clock signál. Takže byli přidány potřebné knihovny. Na obrázcích 4.6, 4.7 a 4.8 jsou představené ty vygenerované inicializační funkce.

```
static void MX_GPIO_Init(void)
{
    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOF_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();
}
```

Obrázek 4.6: Přivedení clock signálu GPIO

Na obrázku 4.6 je vidět, zapnutí hodinového signálu na portech MCU. Port microcontrolleru je to určitá skupina pinů. Díky přivedení clock signálu na port budou aktivovány všechny piny toho portu.

Na obrázku 4.7 je uvedená funkce, která provádí inicializaci I2C sběrnice. Prvním příkazem se dostane jednoznačný název, dále se nastaví hodnoty I2C_Timing registru, zvolí se adresa MCU, nastaví se mode adresování na 7-mi bitové adresy, vypne

```

/* I2C1 init function */
static void MX_I2C1_Init(void)
{
    hi2c1.Instance = I2C1;
    hi2c1.Init.Timing = 0x2000090E;
    hi2c1.Init.OwnAddress1 = 0;
    hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
    hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
    hi2c1.Init.OwnAddress2 = 0;
    hi2c1.Init.OwnAddress2Masks = I2C_OA2_NOMASK;
    hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
    hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
    if (HAL_I2C_Init(&hi2c1) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    /**Configure Analogue filter
    */
    if (HAL_I2CEx_ConfigAnalogFilter(&hi2c1, I2C_ANALOGFILTER_ENABLE) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    /**Configure Digital filter
    */
    if (HAL_I2CEx_ConfigDigitalFilter(&hi2c1, 0) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
}

```

Obrázek 4.7: Inicializace I2C sběrnice

```

/* Private variables -----*/
unsigned char buffer[10];

float CellVoltage1, CellVoltage2, CellVoltage3, CellVoltage4, BattVoltage;
float Vcontroltemperature, Vtemperature, Temperature;

int RegStat7, RegStat6, RegStat5, RegStat4, RegStat3, RegStat2, RegStat1, RegStat0;
int RegCtrlOne7, RegCtrlOne6, RegCtrlOne5, RegCtrlOne4, RegCtrlOne3, RegCtrlOne2, RegCtrlOne1, RegCtrlOne0;
int RegCtrlTwo7, RegCtrlTwo6, RegCtrlTwo5, RegCtrlTwo4, RegCtrlTwo3, RegCtrlTwo2, RegCtrlTwo1, RegCtrlTwo0;
int send;

unsigned int rawCV, rawT, tmp, SCD;

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_I2C1_Init(void);
static void MX_TIM1_Init(void);
static GPIO_InitTypeDef GPIO_InitStructure;

```

Obrázek 4.8: Inicializace proměnných a bufferu

se dvojitě adresovaní, nastavení druhé adresy MCU, vypnutí masky adresovaní a vypnutí dvou doplňujících modu.

Na obrázku 4.8 je představeno nastavení clocků pro systém a I2C sběrnice. Zapínají se dva vnitřní RC oscilátory s kmitočty 8MHz a 48MHz, dále se vypíná PLL násobení a dělení frekvence. Pak je nastavení hodinového signálu pro systém a nastavuje se na 48MHz. I2C sběrnice získává clock signál s frekvencí 8MHz.

Pro komunikace s balancovacím čipem byla potřeba vytvořit buffer, kam se budou zapisovat data před odesláním a během přijímání. Buffer musel být dostatečně velký, aby přijal celý blok dat. Tento blok dat shromáždí v sobě informaci o stavu deseti registrů, do kterých se ukládají naměřené hodnoty napětí. Z důvodu toho, že registry naměřených hodnot se skládají za sebou a mají délku 8-mi bitů, byl vytvořen buffer s 10 položkami o velikosti 8-mi bitů(unsigned char). Inicializace bufferu a proměnných

```

void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct;
    RCC_ClkInitTypeDef RCC_ClkInitStruct;
    RCC_PeriphCLKInitTypeDef PeriphClkInit;

    /**Initializes the CPU, AHB and APB busses clocks
    */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI|RCC_OSCILLATORTYPE_HSI48;
    RCC_OscInitStruct.HSIState = RCC_HSI_ON;
    RCC_OscInitStruct.HSI48State = RCC_HSI48_ON;
    RCC_OscInitStruct.HSICalibrationValue = 16;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    /**Initializes the CPU, AHB and APB busses clocks
    */
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
        |RCC_CLOCKTYPE_PCLK1;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_HSI48;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;

    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_I2C1;
    PeriphClkInit.I2C1ClockSelection = RCC_I2C1CLKSOURCE_HSI;
    if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    /**Configure the SysTick interrupt time
    */
    HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

    /**Configure the SysTick
    */
    HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

    /* SysTick_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
}

```

Obrázek 4.9: Nastavení hodinového signálu systému

je zobrazena na obrázku 4.9.

4.3.2 Nastavení registrů BQ7692006PW

Nastavování registrů balancovacího čipu se provádí způsobem naplnění bufferu správnými hodnotami a vykonáním sekvenci příkazů, které jsou znázorněné na obrázku 4.10.

```

/* Set up CC_EN, CHG_ON and DSG_ON */
buffer[0] = 0x05;
buffer[1] = 0x43;

if(HAL_I2C_Master_Transmit(&hi2c1, 0x18<<1, buffer, 2, 100) == HAL_OK)
{
    // HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_1);
    send++;
}

```

Obrázek 4.10: Sekvence příkazů pro nastavení registrů BQ7692006PW

Postup nastavování je následující:

- do první buňky bufferu nahrát adresu nastavovaného registru
- do druhé buňky bufferu nahrát hexadecimální hodnotu, jak by musel být nastaven registr
- provede se volání funkce pro odesílání bufferu
- zkontroluje se odesílání

Odesílací funkce k vykonání své činnosti potřebuje:

1. jméno sběrnice
2. adresu zařízení, kterému se posílá buffer
3. odkaz na buffer
4. rozměr bufferu, nastavením této hodnoty se dá posílat jenom tu část bufferu, která je potřeba
5. nastavení delay pro přijímání acknowledgu

Hodnota reprezentující nastavení registru má být hexadecimální a ta hodnota se vezme z dalšího výpočtu.

Table 7-8. SYS_CTRL2 (0x05)

BIT	7	6	5	4	3	2	1	0
NAME	DELAY_DIS	CC_EN	CC_ONESHOT	RSVD	RSVD	RSVD	DSG_ON	CHG_ON
RESET	0	0	0	0	0	0	0	0
ACCESS	RW	RW	RW	RW	RW	RW	RW	RW

Obrázek 4.11: SYS_CTRL2 registr

Na obrázku 4.11 je představen SYS_CTRL1 registr. Je potřeba nastavit ho následujícím způsobem: 0 1 0 0 0 0 1 1.

$$01000011_{(2)} = 0x43$$

Takže se nastaví jedničky na potřebných místech, celý registr se vezme jako binární číslo a převede se na hexadecimální hodnotu. Takovým způsobem se provádí nastavení všech potřebných registrů.

4.3.3 Zpětná komunikace

Vyčítání registrů čipu BQ7692006PW bylo naprogramováno z důvodu toho, že vývojové prostředí nemělo debug a byla potřeba kontrolovat nastavení a změny některých registrů například SYS_STAT. Provádělo se to následující sekvencí.

Tato sekvence funguje následujícím způsobem:

- nastaví se adresa místa odkud je potřeba číst a zapíše se do první buňky bufferu
- zavolá se funkce odesílání s použitím nastavené adresy, díky tomu v balančovacím čipu pointer se nastaví na tuto adresu

```

/* Check SYS_STAT register */
#if 1
buffer[0] = 0x00;
if(HAL_I2C_Master_Transmit(&hi2c1, 0x18<<1, buffer, 1, 100) == HAL_OK){}
if(HAL_I2C_Master_Receive(&hi2c1, 0x18<<1, buffer, 1, 100) == HAL_OK)
{
// HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_1);
}

if(buffer[0] & (1 << 0))
{
RegStat0 = 1;
}

if(buffer[0] & (1 << 1))
{
RegStat1 = 1;
error = 1;
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, GPIO_PIN_SET);
}

if(buffer[0] & (1 << 2))
{
RegStat2 = 1;
}

if(buffer[0] & (1 << 3))
{
RegStat3 = 1;
}

if(buffer[0] & (1 << 4))
{
RegStat4 = 1;
}

if(buffer[0] & (1 << 5))
{
RegStat5 = 1;
}

if(buffer[0] & (1 << 6))
{
RegStat6 = 1;
}

if(buffer[0] & (1 << 7))
{
RegStat7 = 1;
}
#endif

```

Obrázek 4.12: Vyčítání SYS_STAT registru čipu BQ7692006PW

- zavolá se příkaz přijímání, který vytvoří pro BQ7692006PW zadání, cílem kterého je odesílání jednoho paketu rozměrem 8 bitu, což je nastavený registr
- ten paket se uloží do bufferu
- pak bitovým maskováním se provede kontrola přijatých bitů a nastaví se příslušné proměnné

Proměnné se dá vypsat pomocí programu STMStudio a získat informaci a stavu čipu.

Kapitola 5

Testování

Po dokončení programování balancer by se měl otestovat. Testovali se ochranné funkce, což jsou odpojení při:

- podbití baterie
- přebíjení baterie
- dosažení maximálního vybíjecího proudu
- zkratu

Aby se začalo testování, při odpojení programátoru a restartovaném balanceru na výstupních svorkách se muselo naměřit napětí baterie, respektive napětí zdroje s připojenou virtuální baterkou.

Za běhu programování čip BQ7692006PW s nastavenými registry byl otestován na ochranu proti podbití. Do registru se zapsala hodnota 0xFF, která byla vypočítaná dříve. Při této hodnotě odpojení zátěži se provádí při dosažení napětí 2,985V. Balancer se napájel ze zdroje s připojenou virtuální baterkou, který dával na výstupu napětí 12V. Při sledování napětí článků programem STMStudio bylo zjištěno, že hodnota těch napětí se nezvídala výše než 2,8V a napětí na vývodech balanceru určených pro připojení zátěži bylo 0V. Tedy test na odpojování při podbití baterie balancer splnil.

Během testování odpojení při nadproudu byla připojena zátěž s proudovým odběrem 0.1A až 0.5A, při které výstup balanceru zůstal sepnutý. Bohužel se v laboratoři nenacházela proměnná zátěž vhodného výkonu pro zjištění prahu vypnutí, teoretická hodnota kterého se rovná 1,6A. Při připojení zátěže s odběrem 4A došlo ke korektnímu vypnutí výstupu, to znamená, že nastavený práh ochrany byl někde mezi 0.5A a 4A. To umožňuje tvrdit, že balancovací čip prakticky splnil test odpojení zátěži při dosažení maximální hodnoty vybíjecího proudu.

Dále byl proveden pokus o nouzové odpojení při zkratu svorek na výstupu. Bohužel, v tomto testu obvod to nezvládl díky vysokým indukčnostem a nízkém průřezu přírodních vodičů. Balancovací čip sice odpojil zkratovaný obvod, ale kvůli vzniku nějaké špičky na napájení BQ7692006PW on ji přijal a poslal dále přímo do napájení ovládacího microcontrolleru. MCU následně se odpálilo. Chybu je nutné detailněji analyzovat, ale toto nebylo v časových možnostech této práce.

Na odpálení MCU testování bylo ukončeno. Ale je potřeba se ještě zmínit o tom, že s vysokou pravděpodobností balancer by splnil test odpojení při přebíti článků. Tento závěr je udělán z toho důvodu, že test odpojení při podbití byl splněn.

Kapitola 6

Závěr

Cílem bakalářské práce bylo vytvořit PCB desku pro ochranu čtyř článkové Li-Ion baterie na základě balancovacího integrovaného obvodu BQ7692006PW od firmy Texas Instruments.

Hlavní cíl této práce byl splněn. První prototyp balanceru na základě uvedeného čipu byl navrhnut, zapájen a otestován na bezpečnostní funkce. Bohužel během programování se nepodařilo splnit doplňující požadavek na odpojení napětí řídicí části PCB desky dlouhým stiskem tlačítka.

Během testování byli získány následující výsledky:

1. test odpojení při podbití baterie byl splněn
2. test odpojení při přebíjení baterie teoretický by mohl být splněn
3. test odpojení při dosažení maximální hodnoty vybíjecího proudu s velkou pravděpodobností je splněn
4. test odpojení při zkratu není splněno

Pro detailnější informaci o testování přejdete do kapitoly 5. Ze získaných výsledků je jasné, že balancer nesplňuje ochranné funkce pro baterie a zatím se nemůže používat pro ochranu Li-Ion baterií. Sice BQ7692006PW při porušení každé z ochranných funkcí prováděl rozpojení napěťového okruhu, ale během testu odpojení při zkratu MCU balanceru bylo zničeno. Důvodem odpálení pravděpodobně stála hardwarová chyba při návrhu (vysoká indukčnost a nízký průřez přívodních vodičů), ale je to předběžný závěr. Chybu je potřeba detailněji analyzovat a zkoumat jiné možnosti.

Předpokládá se pokračování ve vývoje tohoto balanceru pro robota KAMbot. V budoucnu se budou muset opravit drobné chyby, které byly odhaleny při pájení PCB desky a zmíněné v paragrafu 3.2, bude se muset opravit problém odpálení MCU při zkratu výstupních svorek a bude se muset napsat kód pro kompletní ovládání balanceru.

Kapitola 7

Literatura

[1] CROMPTON, T. R. Battery reference book. 3rd ed. Boston: Newnes, 2000. ISBN 978-0-7506-4625-3

[2] POP, Valer. Battery management systems: accurate state-of-charge indication for battery powered applications. Dordrecht: Springer, 2008. ISBN 978-1402069444

[3] CHRUSTALEV, D. A. Akkumulatory. Moscow: Izumrud, 2003. ISBN 5–98131–001–04

[4] Jak aktivně pečovat o Li-Ion články | Vývoj.HW.cz. | Vše o elektronice a programování [online]. Copyright © 1997 [cit. 19.05.2018]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/jak-aktivne-pecovat-o-li-ion-clanky.html>

[5] Zvláštností nabíjení sériových akumulátorů | mobipower.ru [online]. Copyright © 2010 [cit. 19.05.2018]. Dostupné z: <http://www.mobipower.ru/modules.php?name=News&file=article&sid=293>

[6] Balancování náboje baterie zajistí dlouhou pracovní dobu a zvětšuje životnost | rlocman.ru [online]. Copyright © 2009 [cit. 19.05.2018]. Dostupné z: <https://www.rlocman.ru/shem/schematics.html?di=59991>

[7] Často kladené otázky o LiIon člancích | ZAJIC.cz [online] [cit. 19.05.2018]. Dostupné z: <http://www.zajic.cz/nablion/lionfaq.htm>

[8] Používání Li-Ion akumulátorů v praxi | Abeceda baterií a akumulátorů. Články, baterie a akumulátory [online]. Copyright © 2009 [cit. 19.05.2018]. Dostupné z: <http://www.battex.info/hermeticke-akumulatory/li-akumulatory/pouzivani-li-ion-akumulatoru-v-praxi>

[9] Napětí akumulátoru | Abeceda baterií a akumulátorů. Články, baterie a akumulátory [online]. Copyright © 2009 [cit. 19.05.2018]. Dostupné z: <http://www.battex.info/?p=slovnicek-a-pojmy&id=537>.

[10] Paměťový efekt (memory effect) | Abeceda baterií a akumulátorů. Články, baterie a akumulátory [online]. Copyright © 2009 [cit. 19.05.2018]. Dostupné z: <http://www.battex.info/hermeticke-akumulatory/povery-a-myty-o-hermetickych-akumulatorech/pametovy-efekt-memory-effect>

Kapitola 8

Přílohy

Všechní níže popsané přílohy jsou digitální a jsou archivovaný na CD disku, který je částí k odevzdané práci.

Příloha 1 = Celý projekt z AltiumDesigner16 s navrženým balancerem

Příloha 2 = Gerber soubory pro výrobu prototypů

Příloha 3 = Naprogramovaný kód z SW4STM32(Eclipse) s komentáři