České vysoké učení technické v Praze
Fakulta elektrotechnická

Diplomová práce

Automatizovaný systém pro měření parametrů akumulátorových baterií

Praha, 2015 **Autor: Bc. Viktor Ptáček**

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne _____ _____

podpis

Poděkování

Děkuji především vedoucímu této diplomové práce Ing. Pavlu Hrzinovi Phd. za vedení a odbornou pomoc při jejím vytváření, dále děkuji studentovi Bc. Tomáši Reichlovi za pomoc při zapojování rozvaděče. Také bych chtěl poděkovat své přítelkyni Monice Jiráskové za pomoc při grafické úpravě diplomové práce a hlavně bych zde chtěl poděkovat své rodině za podporu při studiích na vysoké škole.

Abstrakt

Diplomová práce se věnuje návrhu a realizaci testovací stanice pro akumulátorové baterie. Důvodem vzniku této práce byla absence testovacího zařízení pro velké baterie na katedře elektrotechnologie. Ve své práci popisuji základní parametry akumulátorové baterie a následně navrhuji zařízení, které dokáže vybíjet a nabíjet baterii dle požadavků laboranta. K řízení takovéto stanice jsem použil řídicí systém cRIO 9004, které mělo za úkol řídit vybíječ, a to generováním PWM signálu s proměnnou střídou, sběr fyzikálních veličin nutných k monitoringu vybíjení a nabíjení baterie a dále starat se o GUI a zálohu naměřených dat jak do vnitřní paměti, tak na vzdálený SQL server. Výsledkem práce je testovací stanice velkých baterií s řídicím softwarem a SQL serverem pro zálohování.

Abstract

This thesis deals with the design and construction of a test station for accumulators. The motivation of this work was the absence of a test station for large capacitance accumulators at the Department of Electrotechnology. There is a description of the basic parameters of accumulators' batteries and design of a device that can discharge and charge the battery according to laborants' specific requirements in my work. For controlling the test station I've used cRIO 9004. Its task is to control the discharger by generating a PWM signal with variable duty cycle, to collect physical quantities, which are necessary for battery charging and discharging monitoring, and to provide GUI and backup of measured data to the internal memory and also to the remote SQL server. The result of my work is the test station for large capacitance accumulators with control software and SQL backup server.

Obsah

Seznam obrázků

Seznam tabulek

Seznam použitých zkratek

VI - Virtual Instrument, Virtuální přístroj je pojem od National instrument. Je to zdrojový kód, který v sobě schovává dataflow program a navíc ještě GUI k tomuto programu.
GUI - Graphical User Interface, uživatelské grafické prostředí
FPGA - Field Programmable Gate Array, programovatelné hradlové pole
V - jednotka napětí volt
DCV - stejnosměrné napětí v jednotkách volt
Ah - označení kapacity - ampér hodina
RT - real-time - aplikace reálného času
IP - internet protokol
MOSFET - Metal Oxide Semiconductor Field Effect Transistor - polem řízený tranzistor
PWM - Pulse Width Modulation - pulzně šířková modulace
UPS - Uninterruptible Power Supply - nepřerušitelný zdroj energie
MHz - megahertz
MB - megabyte
CPU - Central Processing Unit - hlavní procesorová jednotka
VPN - virtual private network - virtuální privátní síť
IFace - InterFace
ISP - Internet service provider - poskytovatel internetového připojení
WAN - Wide Area Network - síť internetu
LAN - Local Area Network - místní síť

Kapitola 1

Úvod

Cílem této práce je navržení a realizace testovacího zařízení pro akumulátorové baterie.

Vznik takové testovací stanice byl již úkolem některých předchozích bakalářských a diplomových prací, ovšem z důvodů nevyhovujících parametrů, především řídicí části a chyb při montáži systému, bylo předchozí zařízení demontováno a zůstal z něj pouze rozvaděč se zakomponovaným nabíjecím zdrojem. Bylo tedy nutné navrhnout zcela novou koncepci testovací stanice.

V první řadě bylo nutné navrhnout a vytvořit novou kabeláž - jak silnoproudu, tak i datové rozvody pro senzory, komunikaci či indikaci provozních stavů.

K řízení mi bylo poskytnuto zařízení NI CRio 9004, jež je z dílny National Instrument. Tento systém by měl plnit u tohoto testovacího zařízení úlohu řídicího prvku, sběru dat, komunikaci s uživatelem přes GUI a zálohování naměřených dat na sql server.

Dále bylo nutné navrhnout způsob měření fyzikálních veličin potřebných pro vybíjení a nabíjení baterií. Konkrétně jde o proud, napětí a teplotu. Dalším úkolem tedy bylo navrhnout a realizovat kabeláž k těmto senzorům a následně je zakomponovat do měřícího systému.

Jako zátěž pro vybíjení baterií měla být použita řízená zátěž, která již vznikla jako předchozí projekt na katedře elektrotechnologie. Bylo nutné k této zátěži navrhnout řízení, otestovat jej a zakomponovat do řídicího systému.

Nakonec celá stanice musela být připojena k sql serveru a bylo zapotřebí provést testovací měření.

Kapitola 2

Popis problému, specifikace cíle

2.1 Popis problému

Na Katedře elektrotechnologie vzniká zařízení pro testování velkých baterií. Vzhledem k tomu, že na této stanici již bylo pro její konstrukci vytvořeno několik bakalářských a diplomových prací, dokumentace k tomuto zařízení je poměrně roztříštěná. Také proto se přistoupilo k demontáži veškerých rozvodů v rozvaděči s vyhlídkou navržení a realizace nové koncepce testovací stanice.

2.2 Specifikace cíle

Cílem této práce je k zařízení navrhnout novou kabeláž a kompozici celé stanice. Vybíjecí a nabíjecí stanice by měla být plně automatizovaná, takže bylo nutné navrhnout a realizovat řízení celé stanice s důrazem na sběr fyzikálních veličin nutných pro monitorovaní stavu systému při nabíjení a vybíjení baterií a s důrazem na řízení proudu baterie při vybíjení. Dále bylo důležité navrhnout i možnosti práce s naměřenými daty, popřípadě realizovat zálohování či migraci dat z tohoto testovacího zařízení.

Kapitola 3

Baterie

Celá práce se zabývá vytvořením testovací stanice baterií. V následující rešerši se budu věnovat olověným bateriím, jejich parametrům a možnostem, jak se baterie dají modelovat.

3.1 Proč olověný akumulátor

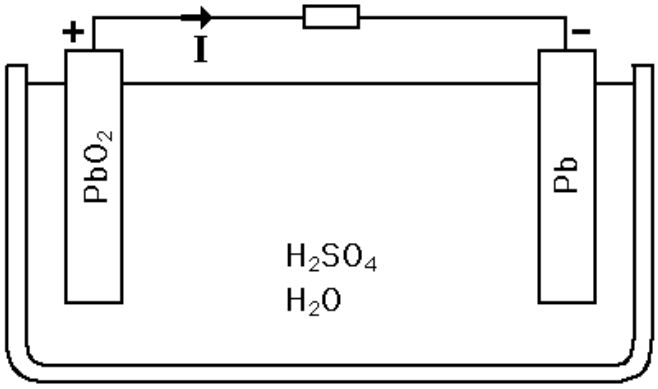
I přesto, že technologie olověného akumulátoru se začala využívat již před více než 150 lety, má stále na trhu s bateriemi velký podíl. Nejčastěji se dnes olověné akumulátory používají jako startovací akumulátory, trakční akumulátory nebo pro UPS systémy. I vzhledem k tomu, že se tato technologie využívá již mnoho let, je dobře popsaná a je k ní dostupných mnoho materiálů. Také baterie samotné jsou velmi dobře dostupné, v každé laboratoři se najde exemplář, na kterém se dají dělat experimenty.

3.2 Historie olověných akumulátorů

Za objevem principu olověného akumulátoru stojí Wilhelm Josef Sinstenden. Tento princip byl publikován v roce 1854. O čtyři roky později sestrojil Gaston Raimond Planté vylepšenou verzi baterie (přenesl laboratorní aparát do „lidské formy“). Olověný akumulátor byl vůbec první komerčně využívaný nabíjecí akumulátor. Problém PB akumulátoru byl v tom, že se nesměl naklánět. To bylo způsobeno jednak tím, že jeho elektrolyt byla kapalná kyselina sýrová a také tím, že při chemické reakci docházelo ke vzniku plynů, které

bylo nutné odvádět. Také bylo zapotřebí občas doplňovat vodu, jelikož při reakci vznikal plynný vodík a kyslík, které se reakcí druhým směrem na vodu zpět nemění. Nakloněním nebo převrácením by tedy mohlo dojít k úniku elektrolytu nebo k zacpání větracích otvorů a mohlo by dojít k výbuchu. Kolem roku 1970 se začala rozšiřovat technologie VRLA, která již nepotřebovala tekutý elektrolyt, resp. byla vyvinuta látka (separátor), ve které byl elektrolyt nasáklý, a tím se zabránilo jeho vytečení. Dále byla v baterii nová technologie, která rekombinovala kyslík a vodík zpět na vodu. V akumulátoru stále zůstal pojistný ventil, který zareaguje na velkou koncentraci plynů. VRLA baterie můžeme rozdělit do dvou skupin - AGM (absorbed glass material) nebo gelové. AGM má elektrolyt nasáklý ve sklolaminátových mikrovláknech, zatímco gelový má elektrolyt ztužený gelem SiOz. [13][15][10]

3.3 Princip olověné baterie



Obrázek 3.1: Schéma olověné baterie, převzato [1]

Nejjednodušší olověná baterie se skládá z anody, kterou tvoří oxid olovičitý a katody, kterou tvoří houbovité olovo. Jako elektrolyt je v baterii použita zředěná (asi 33%) kyselina sírová.

Pro vybíjení platí:

Reakce na anodě:

$$Pb + HSO_4^- \rightarrow PbSO_4 + H^+ + 2e^- [16]$$

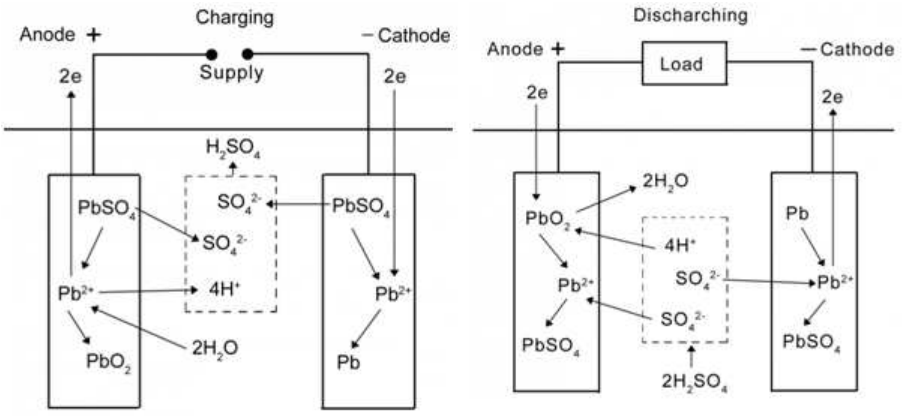
Reakce na katodě:

$$PbO_2 + HSO_4^- + 3H^+ \rightarrow PbSO_4 + 2H_2O[16]$$

Celková reakce:

$$Pb + PbO_2 + 2HSO_4^- + 2H^+ \rightarrow PbSO_4 + 2H_2O + energy$$
$$[16]$$

Pro nabíjení fungují rovnice přesně obráceně.



Obrázek 3.2: Principiální schéma chemické reakce při vybíjení a nabíjení baterie, převzato z [16]

3.4 Parametry olověné baterie

Napětí jednoho článku se pohybuje okolo 2,26V. Omezení pro lead acid baterie je v jejich skladování, měly by se totiž skladovat nabité, aby se tak zabránilo sulfataci. Také existuje omezení ve vybíjení baterie - pakliže dochází k hlubokému vybití, baterie rychle ztrácí kapacitu. Výhodou je velice dobrá cena za Wh.

3.5 Co je to sulfatace?

Sulfatace je proces vzniku stabilních krystalických forem síranu olovnatého, které nemůžeme zpět přeměnit na olovo, oxid olovnatý a kyselinu sírovou.

3.6 Teplota a kapacita baterie

Každá chemická reakce závisí na parametrech prostředí, ve kterém probíhá. Mimo jiné závisí i na teplotě. Jelikož funkce baterie je založena na chemické reakci, i chování baterie bude na teplotě závislé. Při zvyšování teploty se v baterii snižuje vnitřní odpor a reakce v baterii se zrychlují. Navíc se kapacita baterie zvyšuje. V opačném případě, tedy při snižování teploty, dochází ke snižování kapacity, ke zpomalování chemických dějů a zvyšování vnitřního odporu. Tyto vlastnosti baterie způsobují problémy v nabíjení a vybíjení, jelikož s teplotou by se měly měnit i marginální hodnoty napětí článků tak, abychom zachovali baterii v co nejlepší kondici.

3.7 Modelování baterie

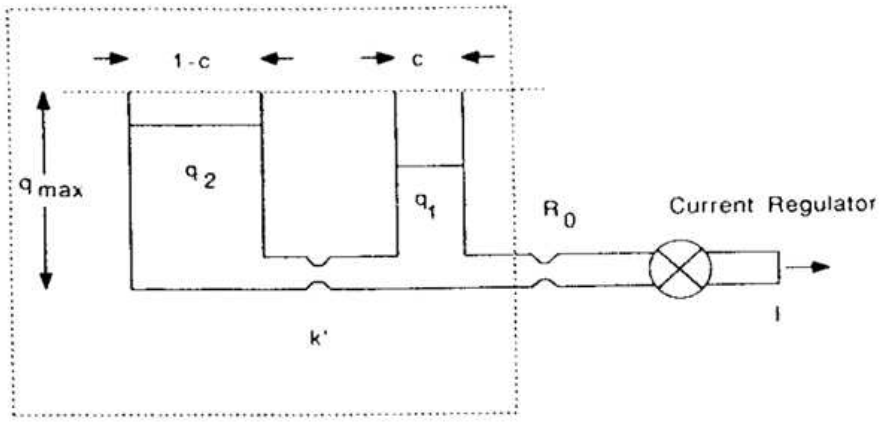
Celé modelování baterie je sice zaměřeno na olověné akumulátory, ale celý systém vybíjení a nabíjení baterie bude použitelný i na ostatní typy akumulátorů. V modelování jako všude, tak i u baterií, existují různé přístupy, různá zjednodušení... Modelování baterií se dá rozdělit do 4 hlavních skupin: první je elektrochemický model, druhý je model s pomocí ekvivalentního elektronického zapojení, dále jsou to modely empirický a kinetický. Všechny čtyři skupiny modelů budou jednotlivě popsány níže.

3.7.1 Elektrochemický model

Elektrochemický model baterie obsahuje popis veškerých chemických reakcí v baterii (nejen popis základních chemických reakcí, který jsem zmínil výše v principu činnosti). K jejich popisu potřebujeme detailně znát vlastnosti materiálů použitých při výrobě.... Tento popis se využívá při vývoji elektrochemických zdrojů.

3.7.2 Kinetický model

Kinetický model je založený na tom, že v baterii jsou uloženy dva různé náboje. Jeden z nich je volný, který je připravený k dodávce proudu do obvodu a druhý je vázaný, který „čeká“ ve formě chemické sloučeniny na chemickou reakci, kterou dojde k uvolnění náboje.



Obrázek 3.3: Schéma pro znázornění principu kinetického modelu

Rychlost přeměny vázaného náboje na volný vyjadřuje konstanta k‘. Konstanta c udává šířku válce s volným nábojem a 1-c udává šířku válce s vázaným nábojem. A platí že (1-c)+(c)=1. R_0 zde nahrazuje vnitřní odpor baterie a „Current regulator“ spotřebič.

Rovnice popisující tento model:

$$\begin{aligned} \frac{dq_1}{dt} &= -I - k'(h_1 - h_2) \\ \frac{dq_2}{dt} &= -k'(h_1 - h_2) \end{aligned}$$

kde

q_1 je volný náboj

q_2 je vázaný náboj

Hladiny h_1 a h_1 se dají vyjádřit jako

$$\begin{aligned} h_1 &= \frac{q_1}{c} \\ h_2 &= \frac{q_2}{c} \end{aligned}$$

Pro zjednodušení se pak dá napsat:

$$k = \frac{k'}{c(1-c)}$$

A potom:

$$\begin{aligned} \frac{dq_1}{dt} &= -I - k(1-c)q_1 + kcq_2 \\ \frac{dq_2}{dt} &= k(1-c)q_1 - kcq_2 \end{aligned}$$

K tomuto modelu je zapotřebí identifikovat pouze 4 konstanty a to q_max, c (rozdělení nábojů), k (rychlost přeměny) a vnitřní odpor.

3.7.3 Veličiny používané v modelování baterií ekvivalentním zapojením

3.7.3.1 Kapacita baterie

Kapacita baterie popisuje, jak velký elektrický náboj je schopna baterie dodat při určitém napětí. Udává se v Ah. Kapacita se většinou udává i s dobou, jakou se tato kapacita má normálně vybíjet. Je to tak proto, že při vybíjení větším proudem dochází ke zmenšení kapacity. K výpočtu potřebné kapacity nám slouží Peukertův zákon

$$C_p = I^k t$$

kde:
C_p je kapacita baterie při vybíjení baterie právě p hodin [Ah],
I je požadovaný proud vybíjení [A],
t je požadovaná doba vybíjení [hod],
k je Peukertova konstanta (u olověných akumulátorů mezi 1,1 až 1,3) [-].

Celý výpočet vypovídá o tom, že když budeme potřebovat vybíjet baterii velkým proudem, je potřeba její kapacitu naddimenzovat nad kapacitu $I \cdot t$. Nominálně se kapacita baterií udává v C_{10}, tj. v proudu, který můžeme brát z baterie po dobu 10 hodin.

3.7.3.2 Stav nabití (state of charge) / hloubka vybití (depth of charge)

Obě veličiny popisují aktuální stav náboje v baterii. Stav nabití představuje poměr aktuálního náboje baterie k nominální kapacitě.

$$SOC = 1 - \frac{Q_e}{C_{10}}$$

Hloubka vybití udává poměr odebraného náboje a kapacity, která odpovídá určitému proudu vybití:

$$DOC = 1 - \frac{Q_e}{C_I}$$

Q_e je náboj odebraný z baterie,
C_I je kapacita baterie při vybíjení proudem I,
C_{10} je nominální kapacita baterie.

3.7.3.3 Samovybíjení

I při nepřipojeném spotřebiči se baterie vybíjí, což je způsobeno vnitřními chemickými reakcemi. Při skladování olověného akumulátoru ve vybitém stavu dochází nejen k samovybíjení, ale i k sulfataci a tím k nevratnému snižování kapacity baterie.

3.7.3.4 Vnitřní odpor baterie

Vnitřní odpor baterie je totéž jako vnitřní odpor zdroje. Čím nižší vnitřní odpor je, tím vyšší proudy můžeme ze zdroje odebírat, aniž by nám kleslo výstupní napětí. Vnitřní odpor baterie má nevýhodu v tom, že se v průběhu vybíjení mění, a tedy závisí na SOC. Při vybíjení akumulátoru roste i jeho vnitřní odpor. Pro výpočet přibližné hodnoty vnitřního odporu se používá vztah

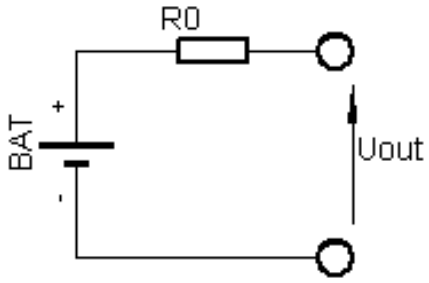
$$R_0 = \frac{U_0 - U_t}{I}$$

U_0 je napětí na prázdno [V],
U_t je napětí při zátěži, kterou teče proud I [V],
I je proud zátěží [A].

3.7.4 Modelování ekvivalentním zapojením

Nejrozšířenějším druhem modelování baterií je modelování pomocí ekvivalentního zapojení. Tento druh je tak rozšířený už jen proto, že většina uživatelů modelu baterií jsou elektrotechnici, kterým jsou elektronické obvody hodně blízké. Navíc se takové obvody dají ve většině případů popsat jednoduchými diferenciálními rovnicemi.

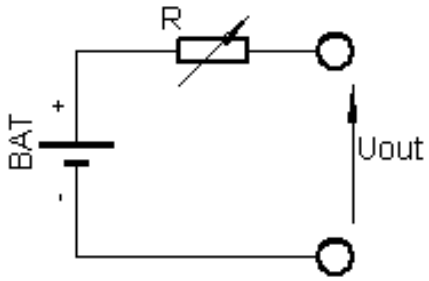
3.7.4.1 Nejjednodušší model baterie



Obrázek 3.4: Schéma modelu - nejjednodušší model baterie

Tento model je lineárním modelem zdroje napětí. Nepopisuje vůbec žádnou dynamiku baterie. Také vnitřní odpor R0 je tu naznačen jako konstantní. Tento model se většinou používá tam, kde nám nevadí změny vzniklé vybíjením baterie.

3.7.4.2 Vylepšený nejjednodušší model



Obrázek 3.5: Schéma modelu - Vylepšený nejjednodušší model

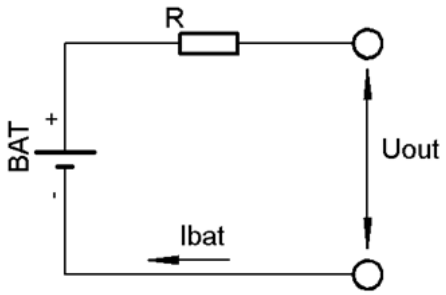
Jediným vylepšením tohoto zapojení je proměnný vnitřní odpor baterie. Tento odpor pak odpovídá vztahu

$$R = \frac{R_0}{SOC^a} \text{ [2]}$$

kde
R je vypočítaný vnitřní odpor baterie,
R_0 je vnitřní odpor plně nabité baterie,
a je konstanta.

Tuto definici jsem získal z materiálu [2], ovšem zde se odkazují na další literaturu, která v dnešní době není dostupná. Takže dle zdroje se nedá určit, jakým způsobem se dá konstanta a odhadnout. Pokud je vztah správný a platí, pak by mělo stačit změřit a vypočítat vnitřní odpor na začátku vybíjení a následně na konci. Z těchto hodnot jsme schopni určit hodnotu a. Čím více hodnot vnitřního odporu při určitém SOC změříme, tím lépe odhadneme konstantu a.

3.7.4.3 Poslední varianta nejjednoduššího modelu



Obrázek 3.6: Schéma modelu - Poslední varianta nejjednoduššího modelu

Celé modelování je založeno na vyjádření výstupního napětí:

$$U_{out} = U_{bat} - R \cdot I_{bat}[2]$$

kde U_{bat} i R jsou závislé na DOC. Teď už stačí říct, v jakém tvaru chceme závislost na DOC - zda chceme lineární ve tvaru:

$$U_{bat} = a_0 + b_0 \cdot DOC[2]$$
$$R_i = a_1 + b_1 \cdot DOC[2]$$

A vnitřní odpor je určen vztahem

$$R_i = \frac{U_0 - U_{term}}{I_{bat}}[2]$$

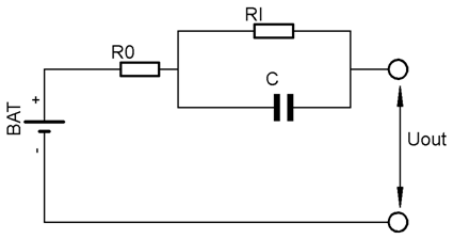
Nebo se dá závislost hledat v polynomiálním tvaru n-tého řádu.

$$U_{bat} = a_0 + a_1 \cdot SOC + a_2 \cdot SOC^2 + a_3 \cdot SOC^3 + a_4 \cdot SOC^4 + a_5 \cdot SOC^5 + ... + a_n \cdot SOC^n[2]$$
$$R = b_0 + b_1 \cdot SOC + b_2 \cdot SOC^2 + b_3 \cdot SOC^3 + b_4 \cdot SOC^4 + b_5 \cdot SOC^5 + ... + b_n \cdot SOC^n[2]$$

Pro identifikaci musíme provést podobné měření jako v předchozím případě, ovšem zde už potřebujeme více hodnot, kterými budeme následně prokládat přímku v lineárním případě, nebo polynom v druhém případě. Výhodou těchto dvou modelů pro identifikaci je, že bychom mohli použít stejné měření a mít dva různé modely.

3.7.4.4 Theveninův model baterie

Je nejčastěji používaným modelem baterie. Na rozdíl od předchozích již vystihuje dynamiku baterie.



Obrázek 3.7: Schéma modelu - Theveninův model

Kde:
BAT je zdroj napětí o hodnotě napětí na prázdno [V],
$R0$ je vnitřní odpor baterie [Ω],
RI je odpor mezi elektrolytem a elektrodami [Ω],
C je aktuální kapacita [Ah].

Všechny tyto parametry se dají určit jako konstanty přibližně, nebo změřit. Ovšem všechny tyto parametry by měly být závislé na SOC.

3.7.4.5 Dynamický empirický model

V literatuře [11] je uveden dynamický empirický model. Bohužel odkaz na další literaturu nefunguje. Celý model baterie je popsán rovnicí

$$U_{out} = U_0 - (R_I + \frac{K}{SOC})I_{bat} \text{ [11]}$$

Kde
U_{out} je výstupní napětí [V],
U_0 je napětí na prázdno [V],
R_I je odpor mezi elektrolytem a elektrodami [Ω],

K je polarizační konstanta, typicky 0,1Ω,
I_{bat} je vybíjecí proud baterie, [A]
SOC je „state of charge“ – stav nabití.

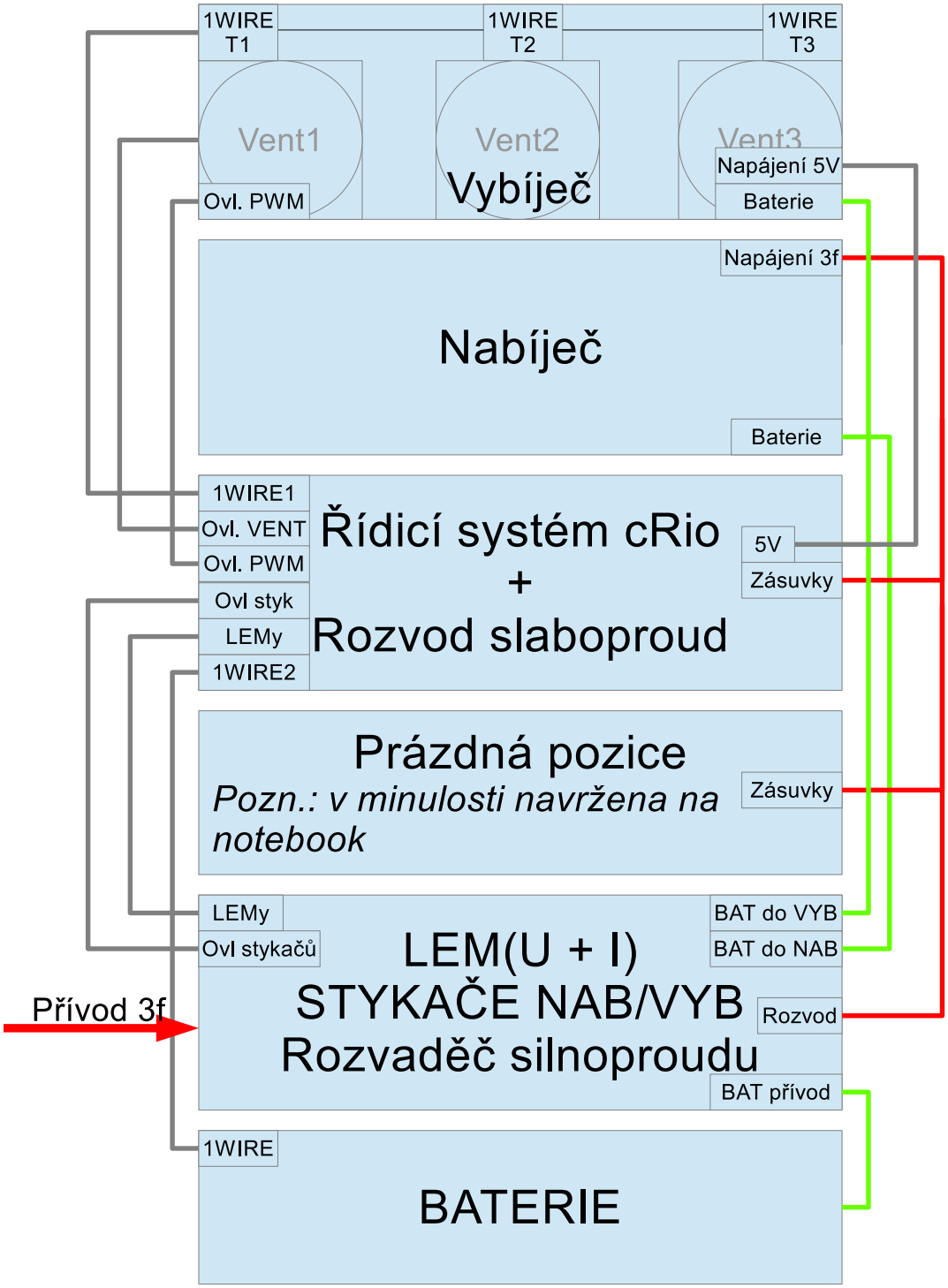
Kapitola 4

Náhled na řešení

4.1 Schéma soustavy a jeho popis

Na počátku návrhu jsem měl k dispozici rozvaděč se zamontovaným zdrojem pro nabíjení. Na tomto rozvaděči jsem si navrhl základní koncepci, kterou můžete vidět na následující straně. Červenou barvou je ve schématu zakreslený rozvod silnoproudu a zeleně je naznačené připojení baterie. Na základě takového návrhu koncepce systému následně vzniklo celkové schéma soustavy (viz příloha), které je rozdělené dle umístění prvků v rozvaděči.

Celý systém má fungovat následovně: Proces se bude provádět na základě vstupních dat od laboranta, která budou obsahovat časové intervaly a informaci, zda se má baterie v daném časovém intervalu vybíjet nebo nabíjet. Pokud by měla být baterie vybíjena, je potřeba dále uvést, jaký proud má být odebírán z baterie. Pokud má být nabíjena, je nutné nastavit na zdroji pro nabíjení takové parametry, aby testu vyhovovaly. Pakliže tyto všechny informace předáme řídicímu systému (popřípadě nabíječi), řídicí systém spustí testování baterie a jeho výstupem budou vzorkované veličiny - napětí baterie, proud baterií, teplota baterie.



Obrázek 4.1: Kompozice systému

4.1.1 Měření proudu LEM sondou

Vzhledem k tomu, že je nutné měřit proud, který jde z baterie, v rozsahu 0 až 150 A, bylo nutné zvolit také adekvátní čidlo. Pro měření takových proudů se používají buď bezkontaktní čidla na základě Hallovy sondy, nebo transformátory pro měření proudu. Nespornou výhodou pro zvolení bezkontaktního měření je galvanické oddělení měřeného objektu s měřícím systémem, a tedy zanedbatelná pravděpodobnost zničení měřícího systému.

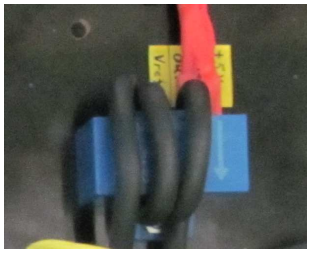
V našich podmínkách jsem zvolil čidlo LEM HASS 500-S. Toto čidlo má rozsah větší než je potřebný rozsah. Rozsah čidla jsem snížil a přesnost zvýšil tím, že jsem měřený drát protáhl 3krát LEM sondou.

Tabulka 4.1: Parametr čidla LEM HASS 500-S

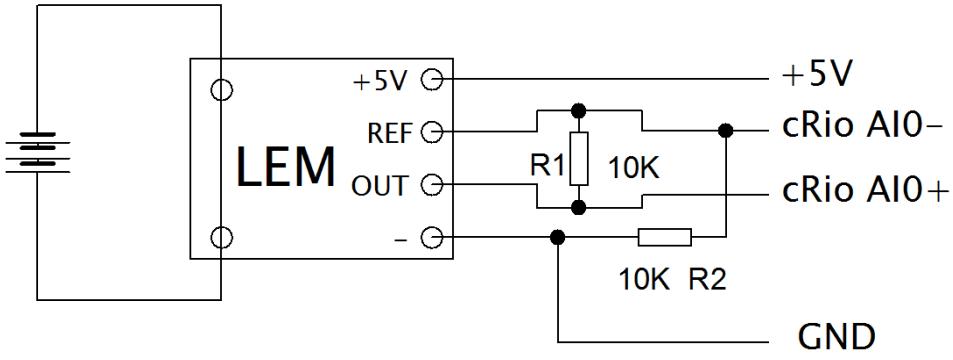
Parametr	Hodnota
Rozsah pro DC proud	± 500A
Výstup	± 2,5V
Napájení	5V
Přesnost	± 1 %

Tabulka 4.2: Nastavené a spočítané parametry čidla LEM HASS 500-S v naší aplikaci

Parametr	Hodnota
Převodní konstanta	273,008
Offset	-0,601562 A
Převodní konstanta pro 3x protáhlý kabel	± 2,5V



Obrázek 4.2: Uchycení LEM sondy v rozvaděči



Obrázek 4.3: Zapojení proudové LEM sondy

Vzhledem k napájení a zapojení budu mít na vstupních svorkách do cRia hodnoty od -2,5V do +2,5V a výsledný proud se pak dá spočítat jako:
Teoreticky:

$$I = \frac{u}{0{,}625 \cdot 500 \cdot pocet\_protazeni}.$$

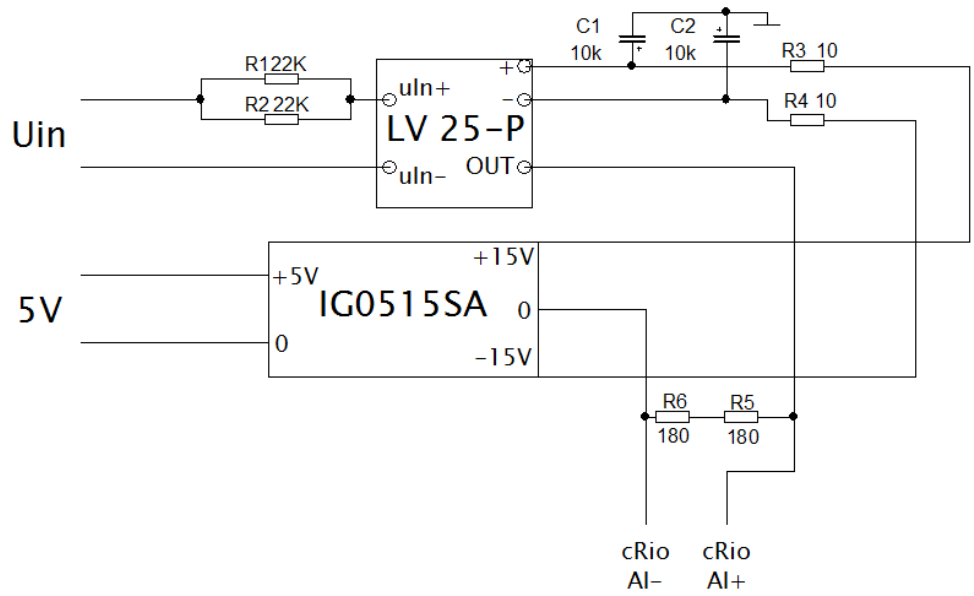
Prakticky:
$I = u \cdot 273,008 - 0,601562.$
Odpor 1 MΩ z AI0- k zemi je dle specifikace cRio modulu.
Odpor 30 kΩ mezi AI0- a AI0+ je zde pro zatížení výstupu LEM sondy.

4.1.2 Měření napětí LEM sondou

Pro měření napětí jsem použil také bezkontaktní čidlo LEM, jen pro napětí. Konkrétně sondu LV25-P. Výhodou použití této sondy je galvanické oddělení měřeného objektu od měřícího systému.

Tabulka 4.3: Parametr čidla LEM HASS 500-S

Parametr	Hodnota
Nominální vstupní proud	± 10mA
Rozsah vstupních proudů	± 0 - 14 mA
Nominální výstupní proud	25mA
Napájení	± 15V



Obrázek 4.4: Zapojení napěťové LEM sondy

Odpory zapojené paralelně u vstupního napětí (22 kΩ || 22 kΩ) jsou navržené tak, aby se sondou dalo měřit 100V a byl využit co největší rozsah sondy. Odpory (180 Ω & 180 Ω) jsou nastaveny tak, aby se využíval co největší rozsah analogových vstupů (± 10 V). Po sestrojení a nastavení odporů jsem si odměřil kalibrační tabulku a z ní pomocí lineární regrese vypočítal převodní konstantu pro toto čidlo.

Tabulka 4.4: Naměřená tabulka pro kalibraci LEM čidla pro měření napětí

Č. měření	1	2	3	4	5	6	7	8
Vstupní napětí [V]	1,03	5,02	10,1	20,07	40,8	60	80	100
Výstupní napětí [V]	0,0775	0,399	0,808	1,613	3,289	4,841	6,455	8,076

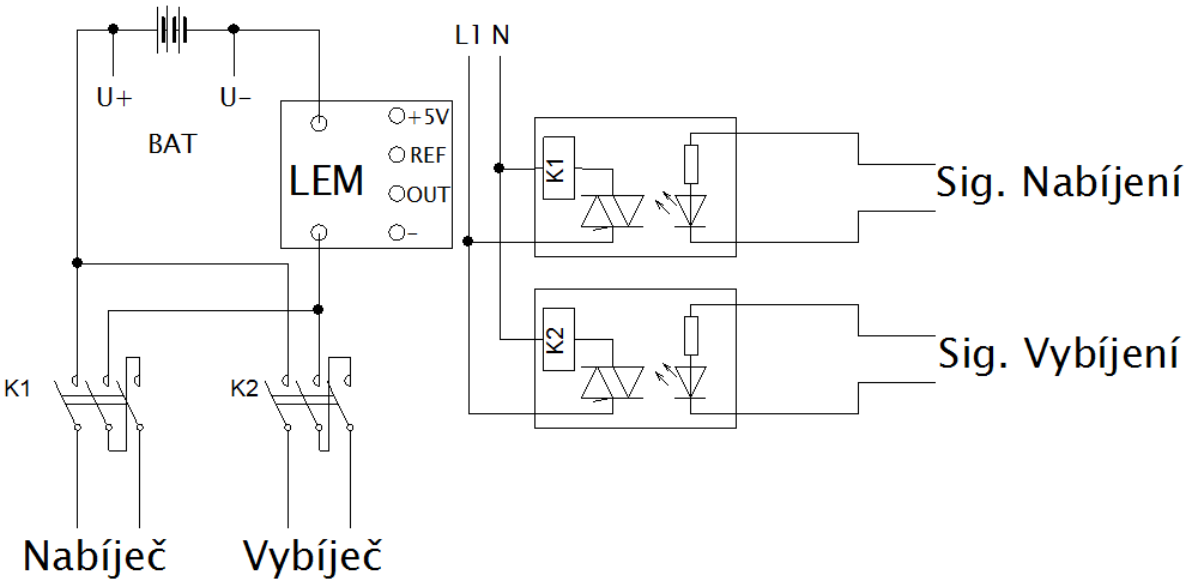
Z naměřených hodnot jsem vypočítal převodní konstantu 12,39, pak pro výpočet napětí platí:

$$U = U_{mer} \cdot 12,39$$

Kde je:
U - hodnota napětí baterie
U_{mer} - hodnota napětí změřená na LEM sondě

4.1.3 Stykače

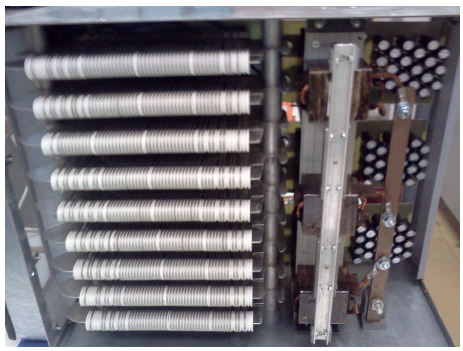
Celý systém umožňuje připnout baterii k nabíječi nebo vybíječi. To je realizováno stykači V140F. Tyto stykače se spínají pomocí síťového napětí 230V. Takovéto napětí není vhodné zavádět do řídicího systému, proto je ke stykači přidáno SSR – solid state relay, a tak se dá řídit pomocí 5V.



Obrázek 4.5: Zapojení stykačů

4.1.4 Řízená zátěž

Na Katedře elektrotechnologie byla jako předcházející projekt vytvořena říditelná zátěž. Tuto říditelnou zátěž tvoří tři identické sekce. Každá z nich je řízena pomocí MOSFET tranzistoru. Jednotlivé sekce jsou ovládány pomocí otevírání a zavírání tranzistoru, v mé aplikaci jsou ovládány PWM signálem. Celý systém je navržen pro maximální proud 150 A. Vzhledem k velkému proudu je měnič osazen i chlazením realizovaným pomocí 3 ventilátorů. Více k řízené zátěži je popsáno v kapitole Řízená zátěž.



Obrázek 4.6: Fotografie vybíječe

4.1.5 Teplotní čidla

K měření teploty bylo zvoleno čidlo Dallas DS18B20. Toto čidlo komunikuje po sběrnici 1wire. Více o tomto čidle je popsáno v kapitole Měření teploty.

4.1.6 Řídicí jednotka

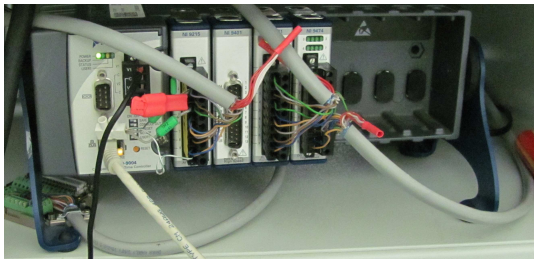
Pro řízení celé soustavy byl vybrán regulátor NI cRio 9004. Jedná se o modulární systém. Centrální jednotka neobsahuje žádný vstup ani výstup, ale musí se do ní dodat měřící a řídící karty. Jako měřící a řídící karty jsem zvolil:

Tabulka 4.5: Seznam použitých měřících a řídících karet do cRia

Zvolená karta	Typ a počet I/O
NI 9215	4 diferenční analogové vstupy, rozsah $\pm 10V$, 16 bit převodník
NI 9401	8 vstupně výstupních kanálů s TTL logikou
NI 9481	4 reléové výstupy
NI 9474	8 digitálních výstupů, maximální rychlost $1\mu s$

Reléové výstupy jsou použity ke spínaní stykačů (2x) a k ovládání větrání na měniči (1x).
Analogové vstupy jsou zapotřebí ke čtení napětí a proudu z LEM čidel (2x).
Vstupně výstupní kartu využívám ke komunikaci s teplotním čidlem po 1-wire sběrnici (4x).

Výstupní kartu využívám k ovládání PWM signálu pro řízení měniče (3x).



Obrázek 4.7: Fotografie řídícího systému osazeného IO kartami

Více informací o tomto regulátoru je popsáno v kapitole Řídicí systém.

4.1.7 Nabíječ Eprona

Nabíječ Eprona HFM-A 80/100 v této testovací stanici působí jako samostatná jednotka. Po nastudování dokumentace a kontaktu s výrobcem jsem se utvrdil v tom, že s jednotkou není možné nijak komunikovat a tedy není možné ji řídit. Tudíž řídicí systém pouze řídí to jestli se má baterie k nabíjecí jednotce připnout a kdy odpojit. Nabíjecí jednotka má třífázový přívod a proto i přívod do testovací stanice musí být třífázový.

Tabulka 4.6: Parametry nabíječe Eprona

Parametr	Hodnota
Vstupní napájení	3x 400V, 50Hz
Rozsah výstupního napětí	0 - 80 V
Rozsah výstupního proudu	0-100A

Kapitola 5

Řídící systém

Jako řídící systém byl zvolen NI cRio 9004 (National Instrument Compact Rio 9004). Tento řídící systém je, na rozdíl od většiny měřících karet, od NI stand-alone, a tedy je v něm integrovaný operační systém a nepotřebuje k řízení nebo monitoringu jakékoliv další zařízení. Jedná se o systém modulární, čili k systému je možné přidat měřící karty dle aktuálních požadavků.

Obrázek 5.1: Fotografie řídícího systému NI cRio 9004, převzato z [17]

V NI cRio běží jako embeded OS LabView Real-time ETS OS. Systém dokáže s okolím komunikovat pomocí protokolů TCP/IP, UDP, Modbus/TCP, Irda a pomocí protokolů pro sériovou linku. Dále systém obsahuje FTP server i http server.

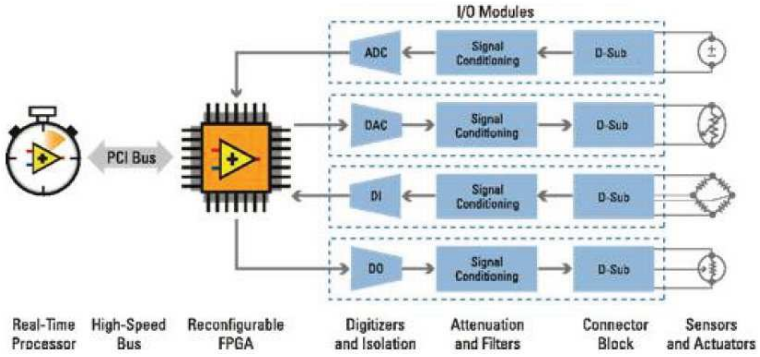
Tabulka 5.1: Parametry přístroje NI cRio 9004 [17]

Název parametru	Hodnota
Rychlost CPU	195 MHz
Permanentní paměť	512 MB
Dočasná paměť	64 MB
Napájení	18VDC – 24VDC

Největší výhodou systému cRio je jeho FPGA procesor. Tento procesor s sebou nese spoustu výhod i nevýhod. Díky architektuře FPGA je možné řídit prvky připojené na vstupně výstupní linky velice rychle. Oddělení jednotky FPGA od procesoru cRia nese výhodu v tom, že při zacyklení uživatelského programu v cRiu to na samotné řízení umístěné v FPGA procesoru nemá žádný vliv.

5.0.8 FPGA procesor

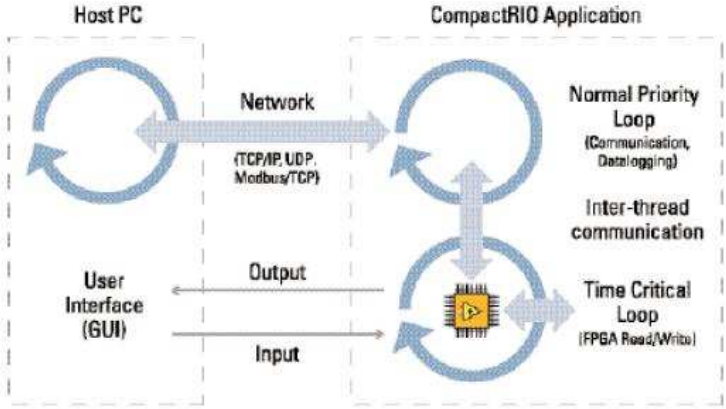
FPGA procesor je programovatelné hradlové pole, kdy na jednom čipu je mnoho logických obvodů a na základě zkompilovaného kódu se tato hradla mezi sebou propojí. Často se takové FPGA procesory programují v jazycích VHDL nebo Verilog, ovšem i LabView má svůj FPGA modul a kompilátor, díky čemuž můžeme v LabView programovat FPGA procesory. FPGA procesor tedy není nic složitého, ovšem je velice obtížné z jakéhokoliv zdrojového kódu propojit tyto hradla v FPGA procesoru tak, aby vše fungovalo. LabView využívá jako kompilátor pro FPGA procesory software Xillinx. Jak jsem již zmínil, celý proces kompilace je velice náročný i na výpočetní výkon, paměť atd. LabView nabízí několik možností, jak můžeme kompilovat FPGA zdrojový kód. První možnost je nejjednodušší, a to nainstalovat na místní disk kompilátor a při nutnosti kompilace tento kompilátor spustit a kompilovat na lokálním počítači. Druhá možnost je využít cloud od National Instrument a na něm zkompilovat svůj kód. Tato alternativa je ovšem placená. Třetí možnost, kterou jsem využil, je využít svůj výkonný počítač a vždy na něj nahrát zdrojový kód a následně ho na něm zkompilovat.



Obrázek 5.2: Architektura Compact Ria, převzato z [18]

5.0.9 Systémové cykly u cRia

Jelikož Compact Rio není klasické PLC, ale obsahuje v sobě FPGA procesor, a navíc obsahuje počítač, v němž běží LabView Real-time ETS, najdeme v cRiu různé systémové cykly. První cyklus je FPGA, který je oddělen od druhého cyklu real-time OS, poslední systémový cyklus je již mimo cRio, a to v počítači, kterým se do řídicího systému připojujeme.



Obrázek 5.3: Systémové cykly Compact Ria, převzato z [17]

Na FPGA procesoru běží části kódů, které jsou nejnáchylnější na časování. V mém případě to je PWM řízení řízené zátěže a komunikace s 1wire čidly. V druhém cyklu běží kódy již méně náročné na čas, a to kódy potřebné pro GUI, pro přepočty hodnot z čidel na "lidské"hodnoty, dále hrubé vzorkování pro zálohování a logování, samotné ukládání

hodnoty, komunikace s sql serverem jak pro účel logování, tak pro účel získání vstupů pro měření.

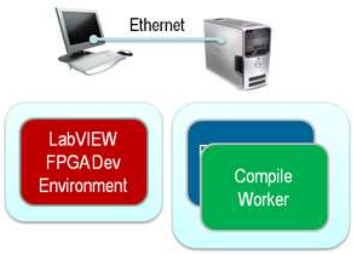
5.1 Jak na svůj kompilační server?

Řešil jsem problém, který spočíval v délce kompilace kódu pro FPGA procesor. Kompilace totiž na laptopu trvala velmi dlouho a po dobu kompilace nebylo možné na laptopu pracovat. Tudíž jsem chtěl kompilaci přesunout jinam než na můj laptop, na kterém jsem po dobu kompilace potřeboval dělat jiné věci.

Celý proces kompilace probíhá tak, že se na počítači spustí kompilace a LabView vytvoří kompilační podklady a následně se tyto podklady nahrají na server. Nejjednodušší možnost je kompilační server pustit na lokálním počítači a nechat kompilaci na lokálním počítači.

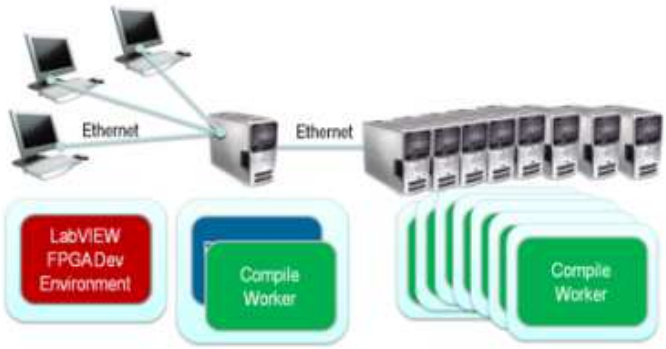
Obrázek 5.4: Architektura kompilace na místním počítači, převzato z [19]

Druhou možností je na vzdálený počítač nainstalovat Compile worker a nahrát kompilační podklady na něj. Tuto možnost jsem při mé práci používal nejčastěji.



Obrázek 5.5: Architektura kompilace na vzdáleném počítači, převzato z [19]

Další možnost je rozložit kompilaci mezi více serverů, kdy jeden z nich je compile server a ostatní od něj přijímají instrukce a jsou compile workers. K tomuto se využívá NI LabVIEW FPGA Compile Farm Toolkit. Taková architektura je vhodná pro velké firmy, kde kompiluje více lidí zároveň a server takto rozděluje úkoly ve frontě mezi několik serverů.



Obrázek 5.6: Architektura kompilace na více serverech, převzato z [19]

5.1.1 Zprovoznění vlastního kompilačního serveru

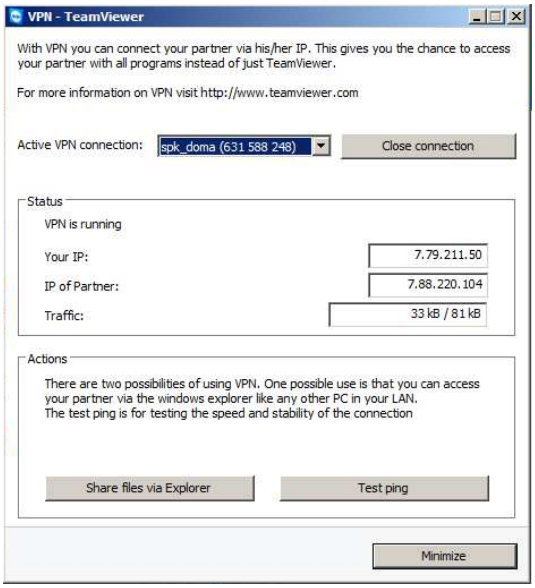
Pokud si chceme udělat vlastní server, potřebujeme na to počítač, na kterém je WIN OS nebo LINUX, na který nainstalujeme Xilinx Compilation Tools. Potom je možné jej použít jako compile server. V mém případě se jednalo o vzdálený server, takže jsem si musel vytvořit VPN tunel.

5.1.1.1 Jak na to?

Na server si nainstalujeme softwarový balík LabVIEW 2014 FPGA Module Xilinx Tools 10.1. Poté musíme povolit uživatelům vzdálené připojení k tomuto serveru. To učiníme v *start ->National Instrument ->FPGA Compile Tools->FPGA Compile Server Configuration.*

Služba compile workeru se nespouští sama, je nutné ji spustit *start ->National Instrument ->FPGA Compile Tools->FPGA Compile Worker.* Poté se již k serveru můžeme připojit a přenechat mu kompilaci. Nastavení kompilačního serveru můžeme udělat přímo ve VI, které chceme kompilovat. Nejdříve si otevřeme *dané VI -> Tools ->options ->FPGA Module ->Connect to a network compile server* a následně vyplníme compile server name, kam napíšeme IP adresu serveru.

Jelikož můj server byl v jiném místě než já, vytvořil jsem si k němu VPN tunel. Použil jsem k tomu produkt TeamViewer, který je pro osobní použití zdarma. Po defaultní instalaci není možné vytvořit VPN tunel, proto je nutné nainstalovat komponentu, která nám to umožní: Otevřeme *TeamViewer ->Extra ->advanced ->show advanced options ->install VPN driver.* Toto se musí udělat na obou počítačích - jak na serveru, tak na klientovi. Následně vyplníme na klientovi Partner ID serveru, zaškrtneme VPN a připojíme se. Následně se otevře okno s informacemi o VPN připojení.



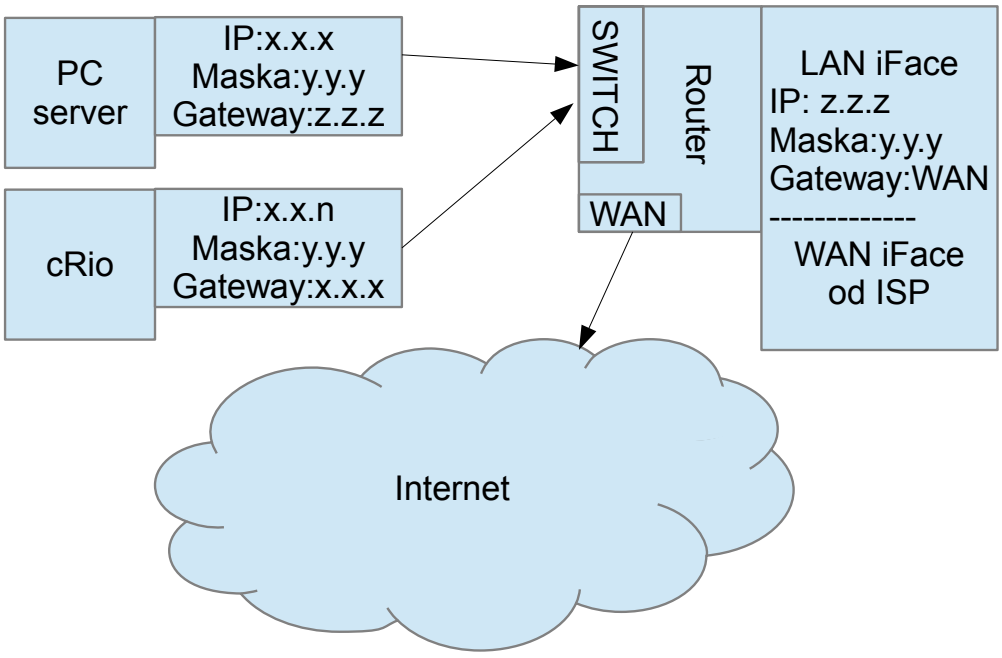
Obrázek 5.7: Okno VPN připojení

IP of partner je tedy adresa, kterou zadáme do LabView jako IP adresu serveru.

Spojení můžeme otestovat pomocí tlačítka Test ping nebo v příkazovém řádku pomocí příkazu ping. Pakliže toto nepůjde, buď se nepovedlo navázat spojení se serverem a můžeme to zkusit znovu, nebo nám v tom brání firewall. Můžeme zkusit vypnout firewall na klientovi i serveru, ale neměli bychom zapomenout firewall vždy opět zapnout.

5.1.1.2 Možnosti připojit se vzdáleně k cRiu

Jedna z možností, jak se vzdáleně připojit ke kontroléru, je, že u něj máme server, na kterém běží veškeré podpůrné nastroje (LabView atd.), a na tento stroj se připojíme pomocí vzdálené plochy (Teamviewer, Windows Remote Desktop. . . .). Další možností je zpřístupnit kontrolér pomocí VPN tunelu. I v takovém případě potřebujeme mít server, který je připojen k internetu a zároveň má připojen do své místní sítě cRio, ale takový server už nemusí mít nainstalován podpůrný software (LabView...). Využijeme jej totiž pouze k síťovému připojení nás k cRiu.



Obrázek 5.8: Topologie pro vzdálené připojení cRia

Nejdůležitější z obrázku je to, že cRio musí být ve stejné síti jako server a navíc musí mít gateway na IP adresu serveru.
Dále je nutné mít na serveru zapnuté routování, což musíme udělat změnou hodnoty v registrech (programem regedit). Je nutné změnit položku IPEnableRouter na hodnotu 1 -

najdeme ji na cestě *HKEY_LOCAL_MACHINE\SYSTEM\ CurrentControlSet\Services\ Tcpip\Parameters*.
Dále je nutné, abychom na klientovi nastavili statickou routu, která pošle pakety pro cRio do VPN sítě. To uděláme z příkazového řádku, a to příkazem

route add IPcRio mask 255.255.255.254 IP_VPN_klienta

Pozn.: Statická routa se z routovací tabulky maže po restartu.
Potom, co vše nastavíme dle výše popsaného a připojíme se VPN tunelem k serveru, měli bychom mít možnost se připojit k cRiu pomocí jeho IP adresy. Zda jsme k cRiu připojeni, můžeme zjistit příkazem ping. Pokud to nepůjde a vše bylo provedeno dle výše popsaného, doporučil bych vypnout firewall na obou počítačích a zkusit to znovu.
POZOR! Pakliže přidáváme routu a IP adresa cRia je stejná jako adresa, která se nachází v klientské lokální síti, může to činit problémy v další komunikaci. V takovém případě doporučuji změnit IP adresu cRia.

5.1.2 Nahrávání softwaru do FPGA

Problém, na který jsem narazil, byl ten, že do FPGA procesoru na cRiu se nedá nahrát více virtuálních přístrojů, ale pouze jeden. Proto jsem všechny mé virtuální přístroje nahrál do jednoho VI a zkompiloval. Kompilace takového VI trvala asi 1 hodinu. Přes dlouhý čas se mi VI podařilo zkompilovat. K nahrání VI do procesoru FPGA je nutné mít validní VI pro FPGA procesory. To je možné ověřit tak, že NI LabView dovolí soubor zkompilovat. Po kompilaci vznikne v záložce build specification příslušný soubor. Po kliknutí na build specification pravým tlačítkem zvolíme properties a zde zkontrolujeme v záložce information zda máme zatržené Run when loaded to FPGA. Pokud toto není zaškrtnuté, je to nutné zaškrtnout a následně znovu sestavit (rebuild) toto VI. Poté opustíme toto okno, stiskneme druhé tlačítko přímo na VI a klikneme na tlačítko download VI to Flash Memory.
Druhý způsob, jak nahrát VI do FPGA procesoru v cRiu, je si v build specification zvolit cestu, kam se má bitfile uložit a sestavit. Následně nahrajeme bitfile z programu : *start->all programs ->National Instrument ->NI Rio ->RIO Device setup*, kde vybereme soubor a zařízení, do kterého ho chceme nahrát a nahrajeme jej do flash memory.

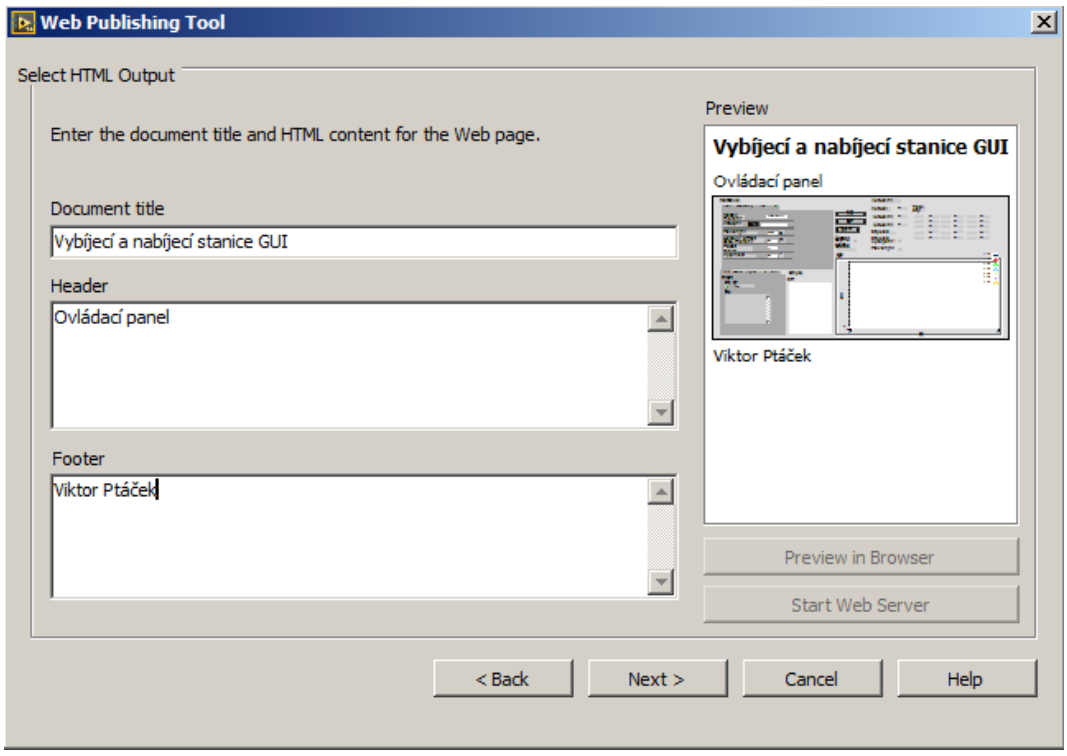
5.1.3 Nahrávání softwaru do cRia

Pokud chceme nahrát software do cRia, a ne do procesoru FPGA, musíme vytvořit build specification. Ta se tvoří kliknutím pravého tlačítka myši na *Build Specifications->New ->Real-Time application*. Následně specifikujeme, jaké VI chceme do cRia nahrát a potvrdíme celé dialogové okno. Následně na tuto Build Specification klikneme pravým tlačítkem myši, zvolíme build a poté stiskneme znovu pravé tlačítko myši a zvolíme položku Run on start up. Tak nahrajeme VI do cRia a spustíme jej.

5.1.4 Remote panel, webová prezentace

Vzhledem k tomu, že v sobě cRio integruje i webový server, je možné se k němu vzdáleně připojit.
První možnost je připojení pomocí Remote panel. Takové připojení nám zpřístupní ovládací prvky VI z cRia, ovšem potřebujeme k němu mít nainstalovaný balík LabView. Druhá možnost je sestavit si webovou prezentaci a následně se připojovat k cRiu pomocí webového prohlížeče a také si takto zpřístupnit ovládací prvky VI z cRia.
Pokud chceme povolit připojení Remote panel, je nutné toto nastavit ve VI. Otevřeme VI z horního menu, vybereme *Tools ->Options ->Web Server* a zde zaškrtneme Enable Remote Panel, následně dole vyplníme, z jaké IP adresy dovolíme VI vidět a z jaké IP adresy dovolíme ovládat prvky ve VI. Také musíme nastavit port, přes který se budeme do cRia připojovat. Po tomto nastavení, sestavení a nahrání takového VI do cRIA se můžeme z LabView připojit k tomuto VI pomocí Remote panel. Připojíme se k němu tak, že VI otevřeme, zvolíme z horního menu *operate ->Connect to Remote Panel*. Zvolíme adresu serveru, jméno VI a zda chceme i ovládat VI. Potom stiskneme connect. Připojování trvá nezvykle dlouho, ale nakonec by se měl zpřístupnit požadovaný panel.

Pro ovládání přes webový server nejdříve musíme vytvořit webovou stránku. To uděláme tak, že otevřeme VI z horního menu, vybereme *Tools ->Web Publishing Tool*. Zde vybereme, z jakého VI má stránka generovat, stiskneme next, vyplníme texty, které se zobrazí na stránce – Nadpis, Popis nad panelem, popis pod panelem. Následně zvolíme umístění, kam se nám má VI vygenerovat. Poté, co webovou stránku vygenerujeme, je nutné ji z počítače přenést do cRia. To uděláme tak, že se do cRia připojíme pomocí ftp protokolu a nakopírujeme vygenerovanou stránku do adresáře c://ni-rt/systém/www/jmeno.html. Následně je tato stránka přístupná pod adresou: http:\\ipcRia\jmeno.html .



Obrázek 5.9: Dialogové okno pro vytváření webové prezentace

Pro ovládání pomocí webové stránky je nutné mít na počítači nainstalován plugin LabVIEW Run-Time Engine.

Kapitola 6

Měření teploty

Jak již vyplývá z modelů baterie, jedna z veličin, která je při nabíjení či vybíjení baterie důležitá, je její teplota. Teplota nám identifikuje stav baterie – při různých teplotách se mění kapacita baterie i její další parametry. Také každá baterie má od výrobce určený rozsah provozních teplot. Dalším důvodem, proč sledovat teplotu baterie, je bezpečnost. Pokud by z nějakého důvodu teplota baterie prudce stoupla nad hranici definovanou kvalifikovanou osobou, je nutné přestat s manipulací (vybíjením nebo nabíjením) baterie.

6.1 Zvolení senzorů teploty

V případě, kdy dostaneme nerozebíratelnou baterii, jediné místo, kde se dá měřit teplota, je na jejím obalu. Zde můžeme zvolit několik bodů, kde budeme teplotu sledovat. Ovšem existují baterie, které jsou složeny z dalších baterií a tvoří pak například trakční baterii. V takovém případě je výhodné měřit teploty na všech bateriích. Ideální je měřit teplotu na jednotlivých článcích zvlášť, tím můžeme identifikovat článek, který může mít defekt. Klasická teplotní čidla mají většinou odporový či napěťový výstup, který nám u našeho PLC obsadí analogový vstup. Tedy kdybychom chtěli 15 čidel, museli bychom PLC osadit 15 analogovými vstupy. Proto jsem se rozhodl zvolit taková teplotní čidla, která umožňují připojení na sběrnici. Jako sběrnici jsem zvolil 1wire od firmy Dallas Semiconductors. Propojení jednotlivých čidel a mastera může být provedeno buď dvěma vodiči (jeden pro obousměrnou komunikaci a druhý jako zem), nebo připojit navíc i napájení 5V na VDD (což je vhodnější pro eliminaci rušení). Bez externího napájení je v teplotním čidle kapacita Cpp, která se nabíjí při logické jedničce na sběrnici a při logické nule napájí

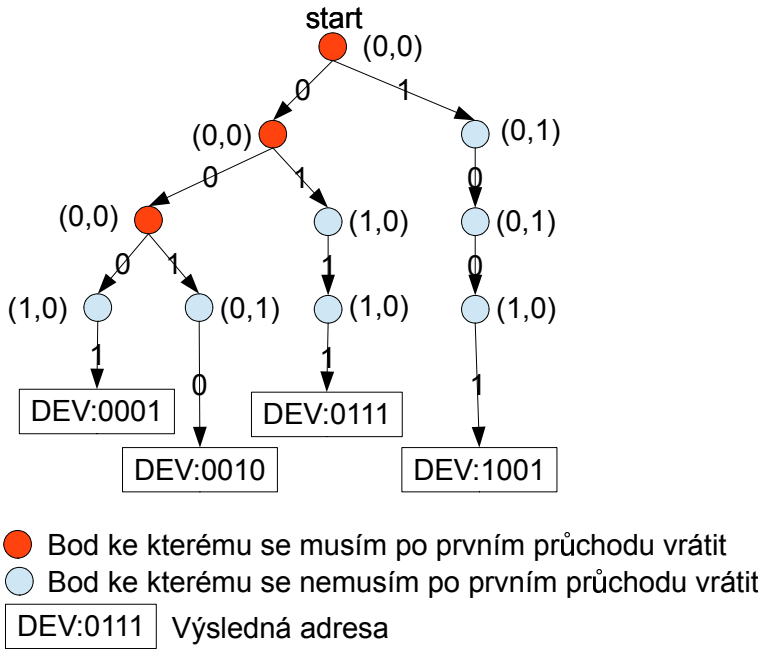
teplotní čidlo. Nakonec jsem zvolil čidlo DS18B20.

Tabulka 6.1: Parametry teplotního čidla DS18B20

Název parametru	Hodnota
Přesnost čidla	±0.5°C
AD převodník	Volitelně 9 až 12 bitů
Pouzdro	TO-92
Sběrnice	1-wire

6.1.1 Vyhledávání čidel na sběrnici

Nejobtížnější úkol při práci s touto sběrnicí je vyhledání jednotlivých slavů na sběrnici. Každé zařízení určené pro 1wire sběrnici má ve své paměti ROM uloženou svoji adresu. Adresa je 64bitová a měla by být jedinečná pro každé zařízení.
Vyhledávání čidla na sběrnici se dá přirovnat k prohledávání grafu do hloubky.
Nejdříve vyšle master do sběrnice příkaz search a zařízení na sběrnici mu odpoví tak, že pošlou nejnižší bit své 64bitové adresy ROM. Jelikož odpovídají všechna zařízení na sběrnici (některé log. 0, některé log.1), výsledkem je logický součin prvních bitů adresy všech zařízení na sběrnici. Poté master požádá o negaci prvního bitu adresy. Tím jsme schopni identifikovat, zda máme na sběrnici všechna zařízení s 0 na konci, nebo pouze zařízení s jedničkou na konci, nebo směs takovýchto zařízení.
Poté master vyšle na sběrnici informaci, že chce komunikovat pouze se zařízeními, která mají na prvním bitu 1 (podle předchozí odezvy, pakliže máme směs zařízení, musíme se nakonec vrátit a projít další větev grafu). Těchto zařízení se ptá, jakou hodnotu mají na druhém bitu. Po odpovědi se ptá, jaká hodnota je negace druhého bitu. Dle toho zjistí, kam se má v grafu vydat a zda se na tento uzel bude muset znovu vrátit.



Obrázek 6.1: Demonstrace vyhledávání čidel na grafu

Začínáme od startu a pokud máme (0,0), musíme se k tomuto bodu vrátit a vyhledat druhou kombinaci. Po nalezení jedné z adres se podíváme do první rozdvojky a pokračujeme druhou kombinací. Až najdeme adresu, před níž nejsou rozdvojky, ke kterým jsme se již nevraceli, máme nalezené všechny adresy.

6.1.2 Naprogramování komunikace

Na stránkách National Instrument již byly zveřejněny základní bloky pro komunikaci s tímto teplotním čidlem. Byly tam vyřešeny věci jako zápis hodnot na sběrnici či čtení z této sběrnice. Ovšem zdrojové kódy byly určeny pro odlišný hardware a nebylo zde prezentováno vyhledávání čidel na sběrnici. Bloku pro vyhledávání se na stránkách NI buď vůbec nevěnovali a problém řešili tak, že využili na každý senzor zvláštní vstup a výstup, nebo jejich řešení nefungovalo. Proto jsem vyvinul vyhledávací algoritmus. Pro vyhledávací algoritmus jsem využil FPGA procesor, který je integrován v cRiu 9004. Při programování FPGA procesoru není možné využít veškeré bloky poskytované systémem LabView, navíc i některé z těchto bloků mají omezenou funkčnost. Největší komplikací, na kterou jsem při práci narazil, byl fakt, že tolikrát, kolikrát jsem nastavil průchod for cyklem, tolikrát se zvýšila velikost programu, což působilo komplikace vzhledem k omezenému místu na FPGA procesoru. Tento problém jsem vyřešil snížením počtu čidel

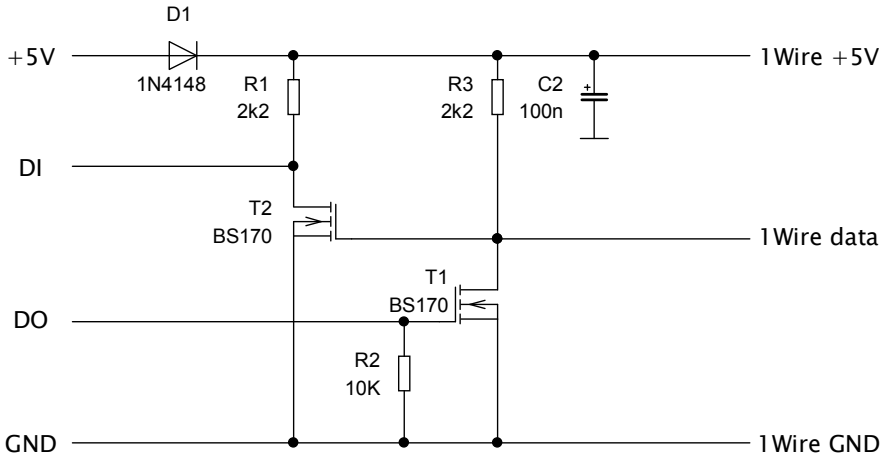
z plánovaných 20 na 15 (každé čidlo = 1 průchod for cyklem).

6.1.3 Dvě oddělené sběrnice 1wire

Na testovací stanici jsou realizovány dvě sběrnice pro měření teploty. První sběrnice má za úkol měřit teplotu baterie. Na této sběrnici může být maximálně 15 čidel. Uživatel má po proskenování sběrnice možnost vybrat si právě teploty, které potřebuje, a také určit jejich název. Tyto možnosti jsou dále popsány v kapitole Popis práce s programem a ukázkové měření. Tyto teploty jsou také využity k hlídání havarijní teploty baterie. Pokud dojde k překročení teploty, dojde k odstavení nabíjení či vybíjení a tato událost bude zaznamenána.
Druhá sběrnice 1wire vede do řízené zátěže. Čidla na této sběrnici jsou umístěna v blízkosti výkonových odporů a v závislosti na nastavené mezi a měřených teplotách se buď zapnou, či vypnou ventilátory určené k chlazení řízené zátěže.

6.1.4 Připojení čidel pro měření teploty

Jak už bylo řečeno, čidla 1wire využívají ke komunikaci jednu linku, která je obousměrná. Ke komunikaci s 1wire teplotními čidly jsem chtěl použít DIO - vstupně výstupní linky, kdy bych si čtení a zápis aktivoval, nebo deaktivoval podle potřeby. Byla mi poskytnuta karta DIO NI 9401. Tato karta sice disponuje vstupně výstupními linkami, ale linky jsou přiřazené do dvou bloků po 4 linkách, kdy vždy celý jeden blok musí být buď vstupní, nebo výstupní. Navíc změna vstupní linky na výstupní trvala určitý čas, což by u komunikace po sběrnici nemuselo být vhodné. Proto jsem přešel k možnosti připojit na datovou sběrnici jak vstup, tak i výstup najednou. K tomuto jsem pro oddělení řídicího systému sestavil jednoduchý driver z tranzistorů.

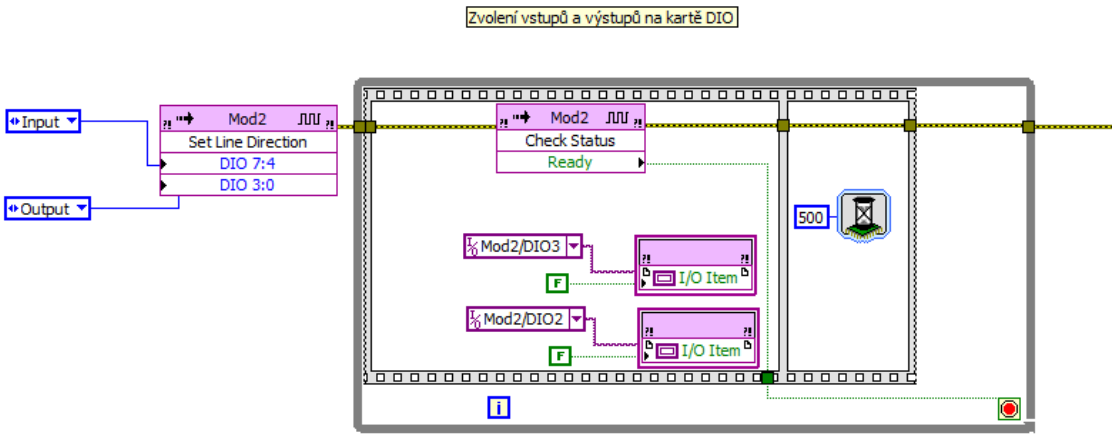


Obrázek 6.2: Driver pro 1wire čidla [20]

Tento driver odděluje sběrnici od řídicího systému, a navíc neguje vstup i výstup.

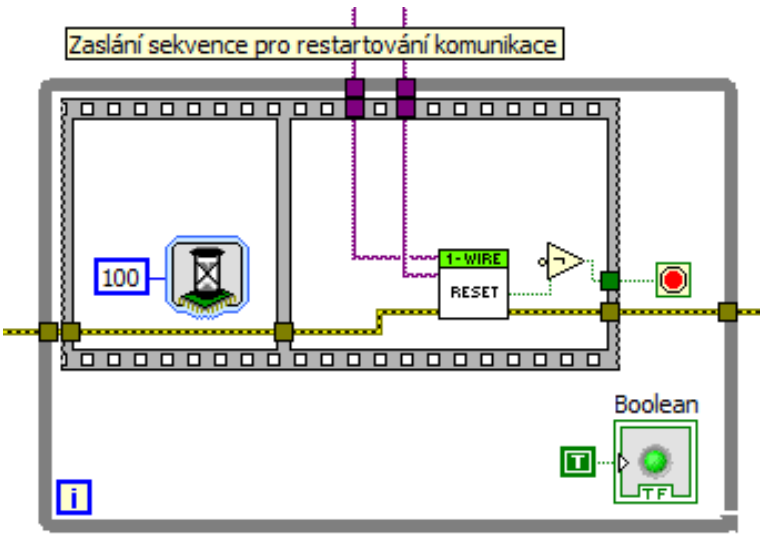
6.1.5 Algoritmus pro komunikaci s teplotním čidlem

Nejdříve tedy přijde inicializace vstupů



Obrázek 6.3: Inicializace IO

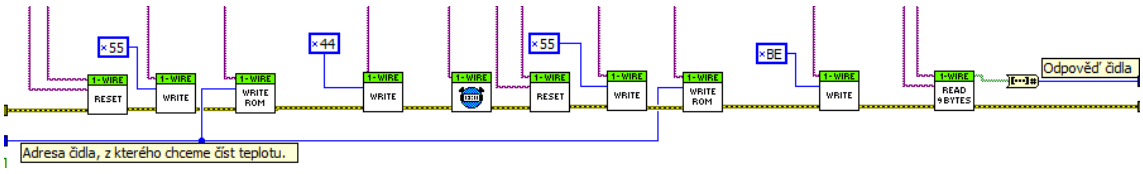
Po inicializaci zašleme na sběrnici sekvenci pro restartování komunikace na sběrnici a počkáme, zda se nám nějaké zařízení ohlásí.



Obrázek 6.4: Zdrojový kód reset sběrnice + čekání na odpověď senzoru

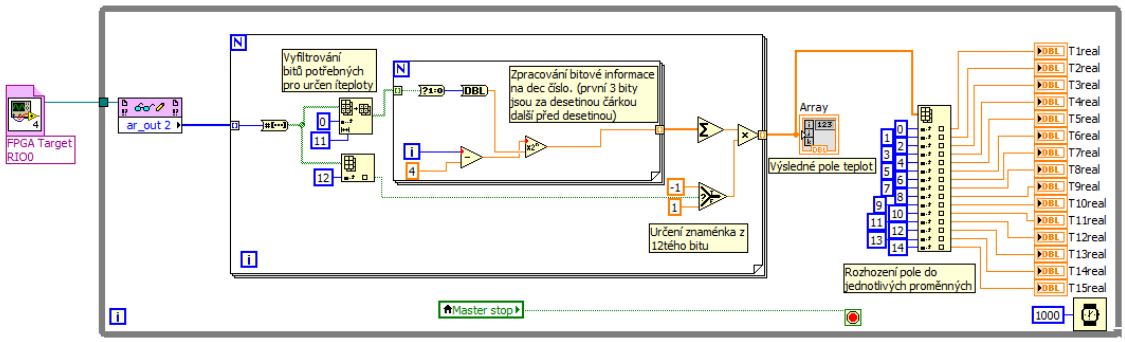
V další části již probíhá hledání na sběrnici. Nejdříve pošleme sekvenci pro restart. Následně zapíšeme na sběrnici xF0h, čímž dáváme zařízením najevo, že vyhledáváme adresy všech zařízení, a následně začínáme hledat.
Celý algoritmus vyhledávání a význam jednotlivých podmínek je popsaný ve zdrojovém kódu.

Poté, co máme veškeré zařízení nalezena a jejich adresy máme v poli, přijde na řadu vyčítání hodnot teploty. Pro vyčtení teploty nejdříve na sběrnici pošleme sekvenci reset a následně sekvenci (0x55h)pro výběr zařízení, se kterým chceme komunikovat. Následně zapíšeme adresu takového zařízení a pošleme mu sekvenci, která říká "změř teplotu". Následně se čeká 750ms pro správný odměr. Pokud máme odměřeno, je nutné přečíst teplotu. Znovu pošleme reset, následně vybereme teplotní čidlo (x44h) a zašleme příkaz pro čidlo, aby nám poslal teplotu. Potom si přečteme 9bytů s informacemi o teplotě. Na obrázku je vidět, že na vstupu je adresa ROM a na výstupu sekvence z čidla. Další vstupy vstupující do tohoto kódu jsou identifikátory vstupu a výstupu cRia (fialově) a err signál (žlutočerná).



Obrázek 6.5: Řetězec instrukcí pro odměr teploty

Převod dat na teploty probíhá výběrem 0. až 11. bitu informace. 3 první bity obsahují informaci před desetinou čárkou a ostatní informaci za desetinou čárkou. Znaménko hodnoty se nakonec určí dle 12. bitu informace. Na obrázku je vidět, jak se na začátku načte pole z VI, které běží na FPGA, a následně se zpracuje.



Obrázek 6.6: Výpočet hodnot z výstupu čidel

Kapitola 7

Datové úložiště

Každé měřící zařízení, které slouží k testování, musí mít architekturu pro zálohu nebo migraci naměřených dat. U této testovací stanice jsem jako úložiště dat zvolil sql server.

7.1 Postavení sql serveru

Počítač, na kterém běží sql server, může mít rozdílné parametry. Pro velké aplikace se používají velké počítače s desítkami procesorů a s terabyty operační paměti. Pro malé aplikace naopak stačí embeded stroje, které mají 700MHz procesor a paměť ve stovkách megabytů. Pro svou práci jsem zvolil počítač s procesorem 2GHz a s pamětí 1024MB. Pouze pro účely zálohování a menší manipulaci s daty je takový stroj i tak značně předimenzovaný.

7.2 Problém při instalaci serveru v síti ČVUT

Jako operační systém jsem zvolil Linux. Konkrétně Debian, který se na malých serverech nasazuje nejčastěji. Při instalaci jsem spoléhal na to, že mi bude sítí ČVUT přidělena IP adresa a balíčky a další software si při instalaci stáhnu z internetu. Ovšem čerstvá instalace Debianu upřednostňuje IPv6, a tak si nejdříve zažádá o tuto IP adresu a pokud ji nedostane, zažádá si o adresu IPv4. V síti ČVUT se dodnes komunikuje po IPv4, ovšem v síti ČVUT je připojeno zařízení, které není oficiálně schválené a přiřazuje IPv6 IP adresy, ale s touto IP adresou se není možné dostat na internet. Proto je důležité na serveru

zakázat IPv6 (povolit jen IPv4). To je možné udělat v souboru /etc/network/interfaces, kde je potřeba napsat:

```
auto eth0
    allow-hotplug eth0
    iface eth0 inet dhcp
```

Po získání správné adresy nebyl problém doinstalovat serverové služby. Ke správě serveru na dálku bylo nutné doinstalovat ssh server (apt-get install ssh). Jelikož datové úložiště mělo být založené na mysql, bylo nutné nainstalovat i mysql server (apt-get instar mysql-server mysql-client). Při instalaci mysql serveru jsem byl vyzván k zadání hesla pro správce databáze (veškerá hesla na serveru jsou „baterky“). V defaultním nastavení mysql serveru není možné spravovat databáze vzdáleně. Je nutné se vždy přihlásit k počítači pomocí ssh a následně se přihlásit na server. Toto jde ovšem změnit tak, že na mysql serveru nastavíme, aby naslouchal na venkovní IP adrese a ne jen na localhost adrese 127.0.0.1. To učiníme v souboru /etc/mysql/my.nc, a zde nastavíme bind-address na adresu mysql serveru, na kterou se chceme připojovat.

Poté se můžeme k databázi přihlásit ze vzdáleného počítače. Ovšem potřebujeme mít na počítači nainstalovaný mysql klienta (tento klient je součástí oficiálního balíku „MySQL Server“). Poté spustíme v příkazovém řádku mysql klienta, který je většinou umístěn na adrese

```
C:\mysql\bin\mysql.exe
```

Pro připojení na vzdálený počítač použijeme příkaz ve tvaru:

```
c:\mysql\bin\mysql.exe -u jmeno_uzivatele -p -h ip_adresa_serveru
```

např.:

```
c:\mysql\bin\mysql.exe -u root -p -h 192.168.1.102
```

Generování dat je založeno na tom, že na serveru je již vytvořena databáze vstup_mereni. Program v řídicím systému si načte tabulky z této databáze a ověří jejich validitu dle počtu sloupců jejich typu. Následně tyto hodnoty přenese do pole, se kterým se dále v programu pracuje.

Podklad pro měření musí být v daném tvaru. Tvar tabulky a datové typy sloupců můžeme poznat z následujícího příkazu, který vytvoří validní tabulku pro vstup měření:

createtable mereni1 (cas int unsigned, proud double, charge boolean, discharge boolean);

První sloupec popisuje čas v sekundách, kdy se mají hodnoty v aktuálním řádku přenést do stanice. Druhý sloupec popisuje hodnotu proudu, kterým chceme baterii zatěžovat. Další dvě hodnoty popisují požadované stavy stykačů. Pokud stavy stykačů budou oba v 1, řídící program toto neuskuteční a oba stykače rozepne, jelikož při této kombinaci stavů stykačů by se baterie připojila zároveň do nabíječe i vybíječe. Ale v řídícím programu je i přepínač, kterým se tato možnost povoluje (ovšem s výstrahou). Tato kombinace stavů stykačů může být žádaná, pokud bychom chtěli změřit zatěžovací charakteristiku nabíječe.

Příklad příkazu pro naplnění tabulky

```
insert into mereni1 values (0, 0, true, false),
(0, 0.1, true, true),
(1, 0.2, false, true),
(2, 1, true, false),
(3, 2, false, true),
(5, 4, true, false),
(80, 5, false, true),
(90, 90, true, false),
(22, 89, true, false),
(34, 7.98, false, true),
(100, 1, true, false),
(120, 2, false, true),
(250, 10, true, false),
(300, 17, false, true),
(340, 54, true, false),
(500, 8, false, true),
(670, 90, true, false),
(790, 9, true, false),
(890, 30, false, true),
```

Tabulka 7.1: Důležité příkazy pro orientaci v mysql databázi

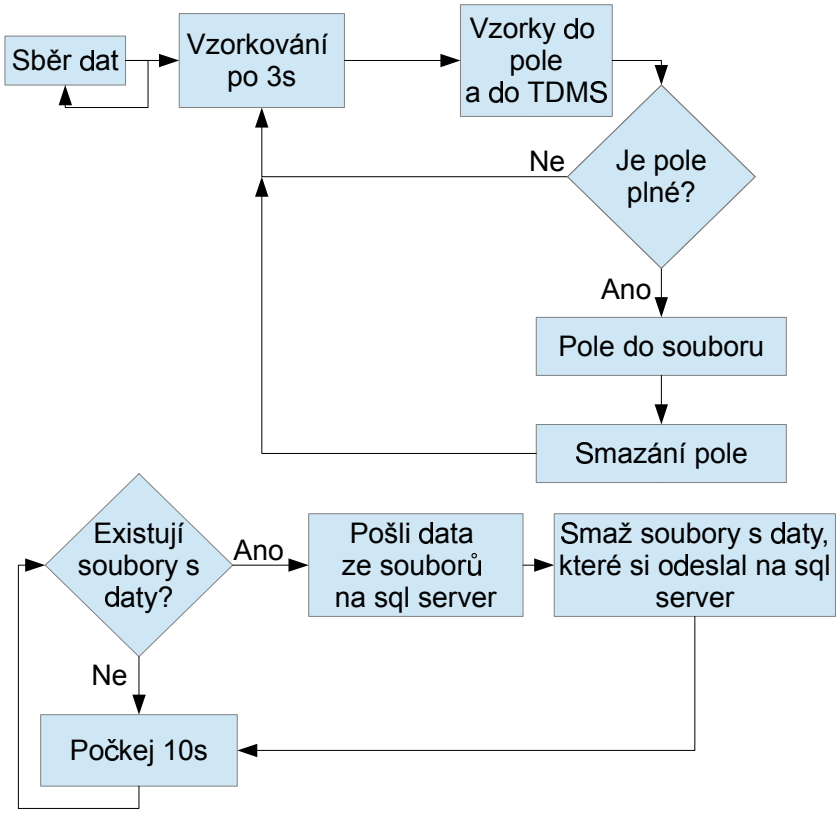
Význam příkazu	Příkaz
Velikost databáze	table_schema "jmeno_databaze", Round(Sum(data_length + index_length) / 1024 / 1024, 1) "velikost databáze v MB"FROM information_schema.tables GROUP BY table_schema;
Zvolení databáze, ve které chceme pracovat	use jmeno_databaze;
Vybrání všech prvků z tabulky	select * from jmeno_tabulky;
Vybrání všech prvků z tabulky a seřazení	select * from jmeno_tabulky order by jmeno_sloupce
Vypsání databází	show databases;
Vypsání tabulek	Show tables;
Vytvoření tabulky	create table mereni1 (jmeno_sloupce datový_typ, jemno_sloupce2 datový_typ,...);
Vytvoření databáze	create database jmeno_databáze
Nastavení práva připojit se k databázi vzdáleně z určité IP	GRANT ALL PRIVILEGES ON databaze.* TO 'user_name'@'IP' WITH GRANT OPTION;
Ukončení klienta	Exit

7.3 Topologie zálohování

Celý proces zálohování je založen na dvou smyčkách.
V první smyčce se sbírají data, když je vzorků dostatek, vezmou se a uloží do souboru. Při sběru dat se průběžně také ukládají data do cRia a to v podobě TDMS souboru.

V druhé smyčce se otevře adresář, kam se soubory s daty ukládají, vezme se první soubor, sestaví se dotaz pro sql server pro uložení těchto dat, následně se tato data nahrají na sql server do předem vytvořené tabulky (vytvořené na začátku měření, a to pod názvem měření). Pokud uložení na sql server proběhlo v pořádku, aktuálně zpracovávaný soubor se smaže a pokračuje se dalším souborem. Pakliže ve složce žádný soubor není, čeká se,

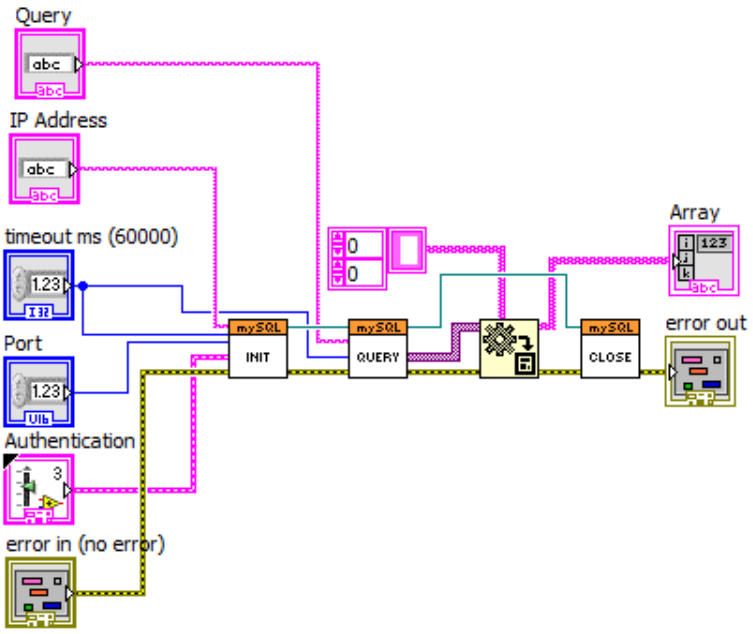
až proces sběru dat uloží další soubor s daty.



Obrázek 7.1: Naznačená topologie zalohování

7.4 Připojení cRia k sql serveru

Připojení k mysql serveru je popsáno na oficiálních stránkách pomocí Database connectivity toolkit. Ten ovšem nepodporuje NI cRio 9004, tudíž bylo nutné najít jinou alternativu. Na stránkách National Instrument v sekci komunita jsem našel Native LabVIEW TCP/IP Connector for mySQL Database. Výhodou tohoto driveru pro mysql je, že funguje i v našem cRio 9004. Jediný problém je v tom, že dokumentace, podle níž byl tento driver vyhotoven, již není k dispozici. Po otestování však vše funguje správně.



Obrázek 7.2: Ukázka VI, které realizuje připojení k mysql serveru z cRia

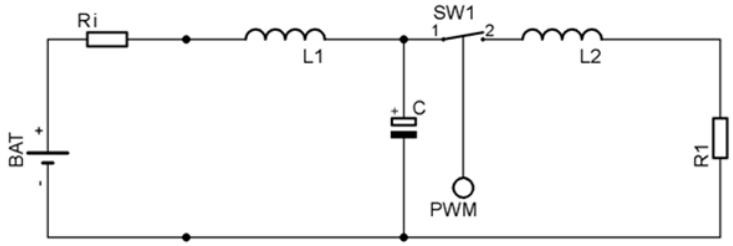
Kapitola 8

Řízená zátěž

DC-DC měniče jsou zapojení, která se globálně používají pro změnu velikosti stejnosměrného napětí nebo proudu. Od klasických členů používaných ke změnám napětí jako jsou diody, odporové děliče... mají tyto měniče nesrovnatelně větší účinnost. Ale kvalita takového měniče a jak se bude chovat záleží na jeho řídicím prvku.
U DC-DC měničů step-down je využito spínání zdroje k zátěži pomocí spínacího prvku ve formě tranzistoru. Jako zátěž zde není použit pouze odpor, ale je zde použit v nějaké formě RLC článek, který přináší při spínání vysokými frekvencemi velice malé zvlnění výsledného DC napětí.
Většina DC-DC měničů je postavena tak, že máme zdroj napětí, který můžeme libovolně zatěžovat, ale je tu spotřebič, který potřebuje určité napětí bez ohledu na to, zda zdroj kolísá či nikoliv. V takovém případě se využívají DC-DC měniče, které toto napětí ze zdroje upravují pro požadavky spotřebiče. V našem případě je ovšem vše naopak. Důležité totiž je, jak se chovají parametry u zdroje, nikoli na straně spotřebiče.

8.1 Naše řízená zátěž

Vybíjecí část testovací soustavy se skládá z vybíječe, který je realizován jako 3 samostatné DC-DC měniče. Tyto DC-DC měniče jsou zapojené paralelně, tj. tak, aby jimi mohl téci co největší proud.



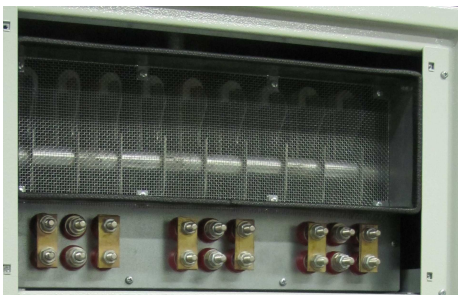
Obrázek 8.1: Ekvivalentní zapojení pro jednu sekci DC-DC měniče

Kde Ri naznačuje vnitřní odpor zdroje, který chceme vybíjet. L1 a C je osazena přímo v měniči a slouží k vyfiltrování proudu tak, aby proud z baterie byl co nejméně zvlněný. Dále je zde naznačený spínací prvek SW1. Ten je realizován v našem případě IGBT tranzistorem. Tento tranzistor je spínaný ze signálu, který generuje řídicí systém. Jako zátěž je zde odpor, který je navinut z odporového drátu. Jelikož je navinut z odporového drátu, má i určitou indukčnost, která je naznačena ve schématu cívkou L2. Vzhledem k maximálním proudům, které mohou procházet spínacím tranzistorem, je i maximální proud, který může měnič odebírat z baterie, omezen. Konkrétně je toto omezení 50A na jednu sekci, tj. celkově 150A pro celý měnič.

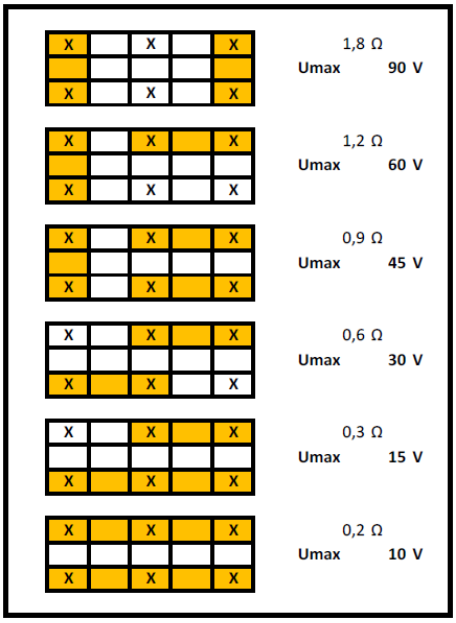
Aby se daly měřit baterie s různým napětím a maximálním proudem 150A, je zde možnost přepínání odporů R1. Toto přepínání odporů je realizováno na jednotlivých sekcích zvlášť.

Tabulka 8.1: Tabulka maximální napětí k nastavenému odporu řízené zátěže

Odpor	Maximální napětí
1,8Ω	90V
1,2Ω	60V
0,9Ω	45V
0,6Ω	30V
0,3Ω	15V
0,2Ω	10V



Obrázek 8.2: Fotografie svorek pro přepínání odporů v řízené zátěži



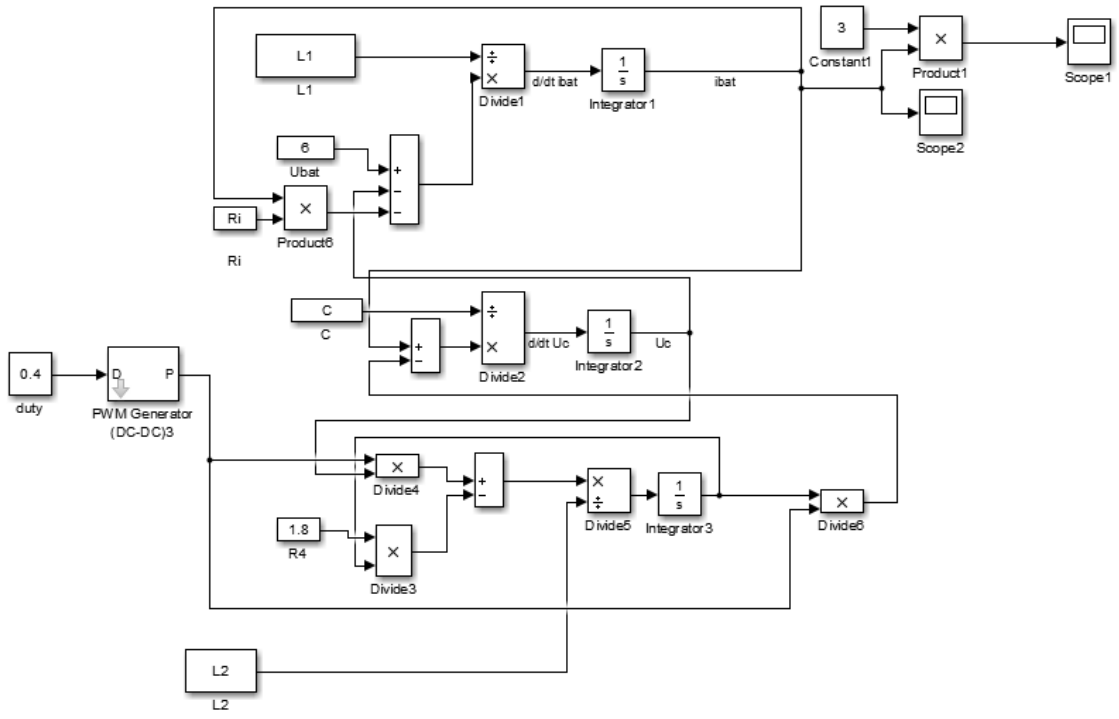
Obrázek 8.3: Tabulka ukazující, jak zapojit svorky na řízené zátěži pro určité odpory

8.2 Modelování řízené zátěže

Model řízené zátěže vznikl pomocí náhradního schématu jedné sekce. Na základě tohoto schématu jsem popsal obvodové rovnice (PWM signál bude nabývat hodnot 1;0):

$$\begin{aligned}\frac{di_{bat}}{dt} &= \frac{u_{bat}}{L_1} - \frac{Ri \cdot i_{bat}}{L_1} - \frac{u_C}{L_1} \\ \frac{du_C}{dt} &= \frac{i_{bat}}{C} - \frac{PWM \cdot i_{L_2}}{C} \\ \frac{di_{L_2}}{dt} &= -\frac{R_1 \cdot i_{L_2}}{L_2} + \frac{u_c \cdot PWM}{L_2}\end{aligned}$$

Dle těchto diferenciálních rovnic jsem vytvořil model v Simulinku. Následně jsem změřil odezvy řízené zátěže a srovnal je s odezvami nasimulovanými v Simulinku. Odezvy se neshodovaly, proto jsem parametry modelu měnil do té doby, než jsem byl s odezvami spokojen. Nakonec mi parametry modelu vyšly následovně:



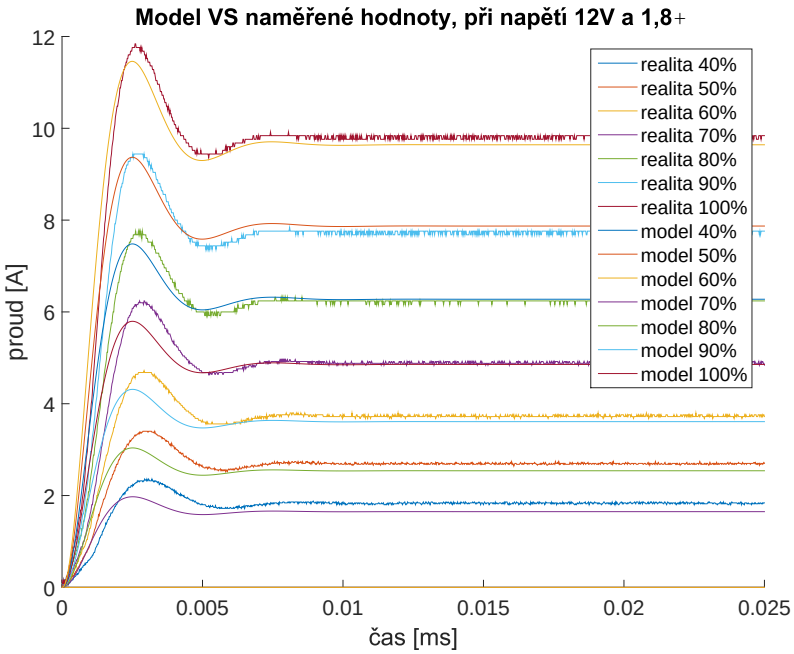
Obrázek 8.4: Model v Simulinku

Tabulka 8.2: Vyladěné parametry modelu

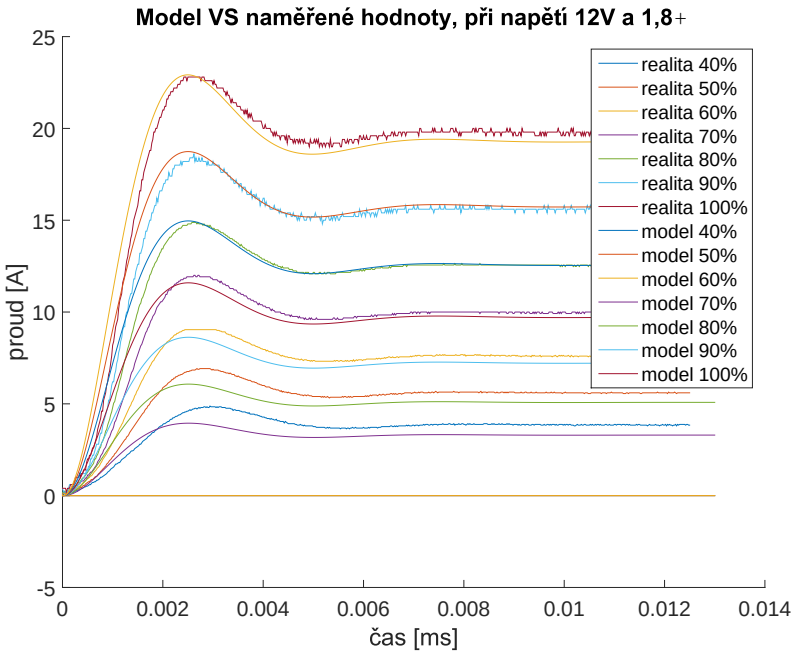
Popis veličiny	hodnota
L1	52uH
L2	není předem určena
R1	1,8Ω
C	9,87mF
Ri	není předem určen - v setinách nebo desetinách ohmů

Tabulka 8.3: Zadané hodnoty reálné řízené zátěže

Popis veličiny	hodnota
L1	32uH
L2	?
R1	1,8Ω
C	9,87mF
Ri	? v setinách nebo desetinách ohmů



Obrázek 8.5: Srovnání odezev na skoky reálný systém VS. model



Obrázek 8.6: Srovnání odezev na skoky reálný systém VS. model

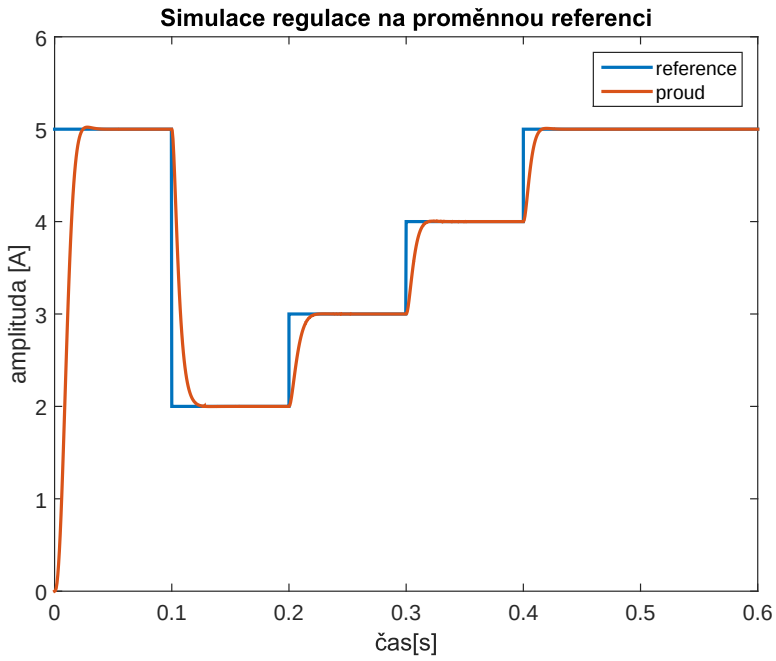
8.2.1 Modelování pomocí průměrné hodnoty

Modelování je založené na tom, že když máme takovou soustavu rovnic závislou na PWM(výše), pak části rovnic, u kterých není prvek PWM, se do chování obvodu promítnou nezávisle na hodnotě PWM. Podíl ostatních veličin na chování obvodu závisí již na signálu PWM. Jelikož signál PWM nám určuje dvoustavově, zda se prvek připojí či nikoliv, a zároveň jeho frekvence je hodně vysoká, můžeme spočítat jeho střídu a následně ji dosadit za parametr PWM. Tím nám vznikne soustava rovnic popisující měnič averrage metodou. Tuto metodu jsem v Simulinku nasimuloval a zjistil, že reakce systému averrage metodou a simulace pomocí PWM byly naprosto stejné.

8.3 Řízení říditelné zátěže

K řízení říditelné zátěže jsem zvolil PID regulátor. Nejdříve jsem toto řízení nasimuloval na modelu pomocí nástroje Simulink a zjistil jsem, že nejlepší kombinací pro PID regulátor je malá P Gain a doladěná I gain. Navíc do PID regulátoru jsem pouštěl proud vztažený

k maximálnímu proudu (vždy hodnotu proudu v procentech vztaženou k maximálnímu proudu pro aktuální U a R). Podle původní představy se dle napětí a nastaveného odporu měl přepočítat vstup do PID regulátoru a všechny další parametry PID regulátoru měly zůstat vždy stejné, což ovšem v praxi fungovalo jen částečně (viz dále v ladění PID regulátoru).

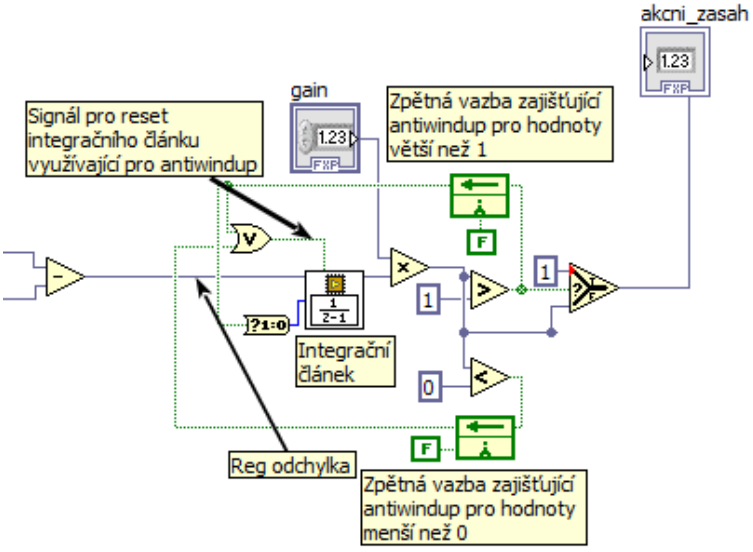


Obrázek 8.7: Graf ze simulace regulace na proměnnou referenci

Po nasazení PID regulátoru na reálnou řízenou zátěž jsem zjistil, že zvýšení P Gain systém poměrně hodně rozkmitává, proto jako regulátor používám pouze vyladěný integrační článek.

8.3.1 Integrační článek antiwindup

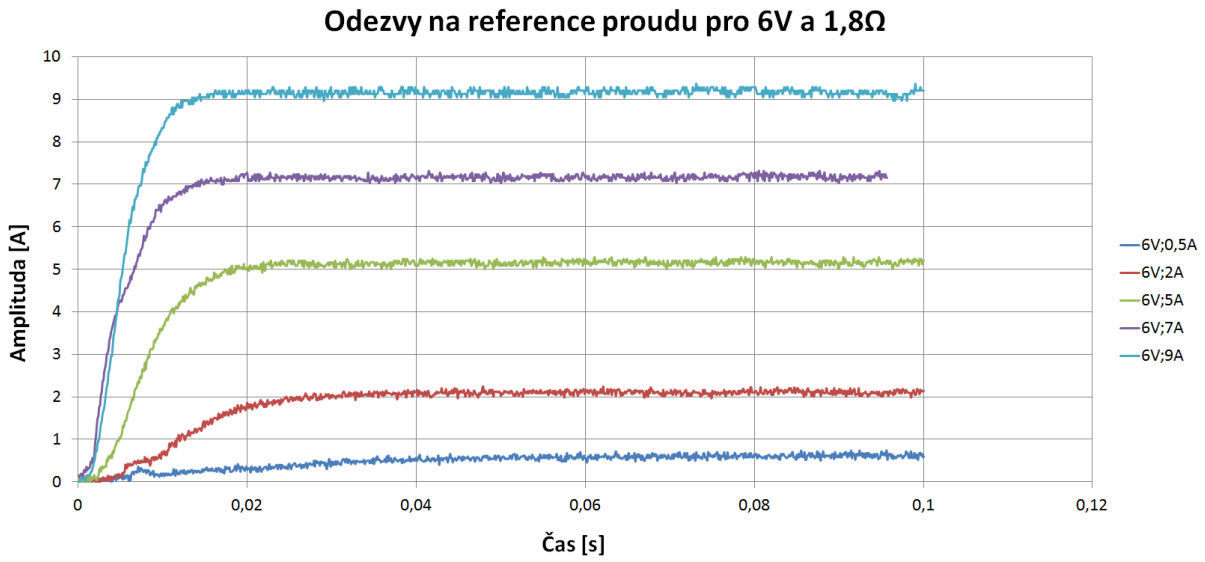
Vzhledem k úspoře paměti v FPGA procesoru a protože nepotřebuji P Gain ani D Gain, nepoužívám PID express VI, nýbrž pouze jednoduchý integrační článek. Zatímco v express VI by si veškeré věci ohledně regulátoru PID řešilo LabView samo, u samostatného integračního článku je nutné věci jako antiwindup udělat zvlášť.



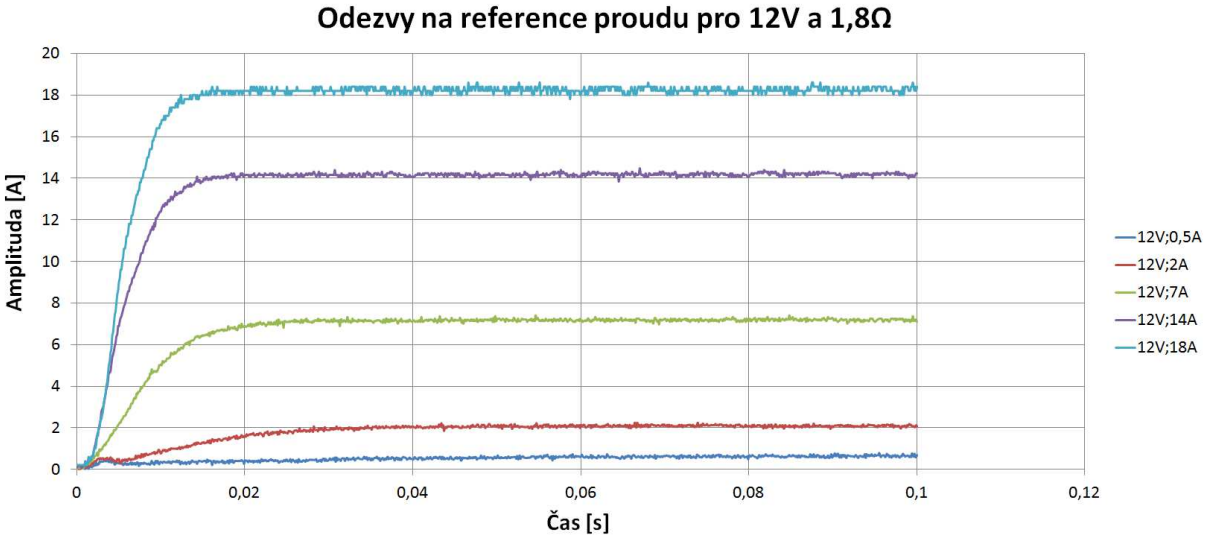
Obrázek 8.8: Část zdrojového kódu regulace znázorňující I regulátor s antiwindup vylepšením

8.3.2 Ladění I článku

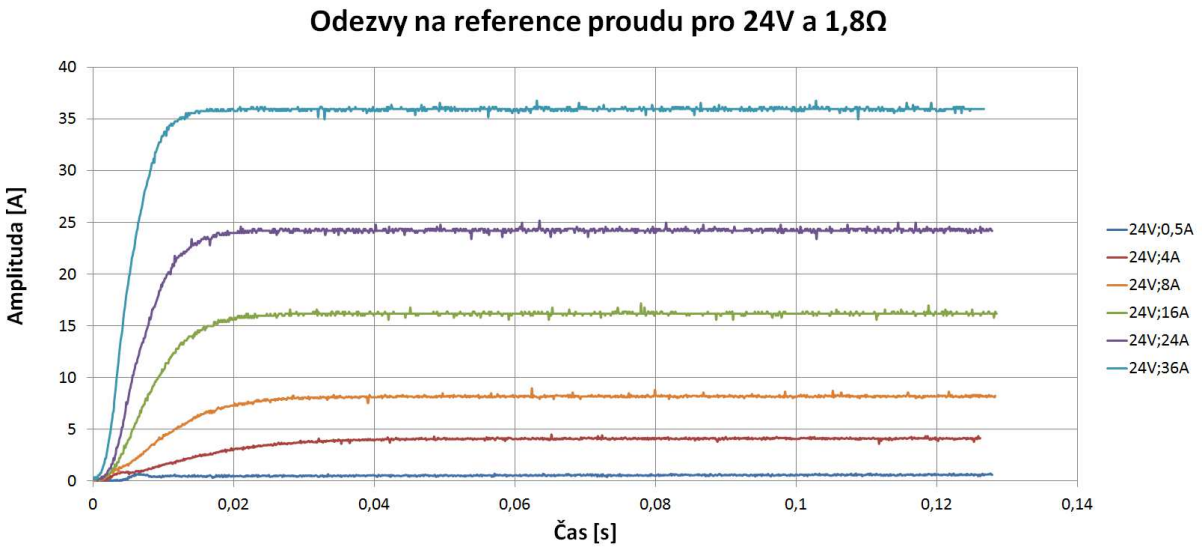
Ladění I regulátoru probíhalo přímo na řízené zátěži, kde jsem sledoval jednotlivé odezvy pro různá napětí a proudy. Také jsem přepínal jednotlivé rozsahy řízené zátěže. Pro měření odezev jsem použil laboratorní zdroj Satron 30V; 60A. Naměřil jsem následující odezvy:



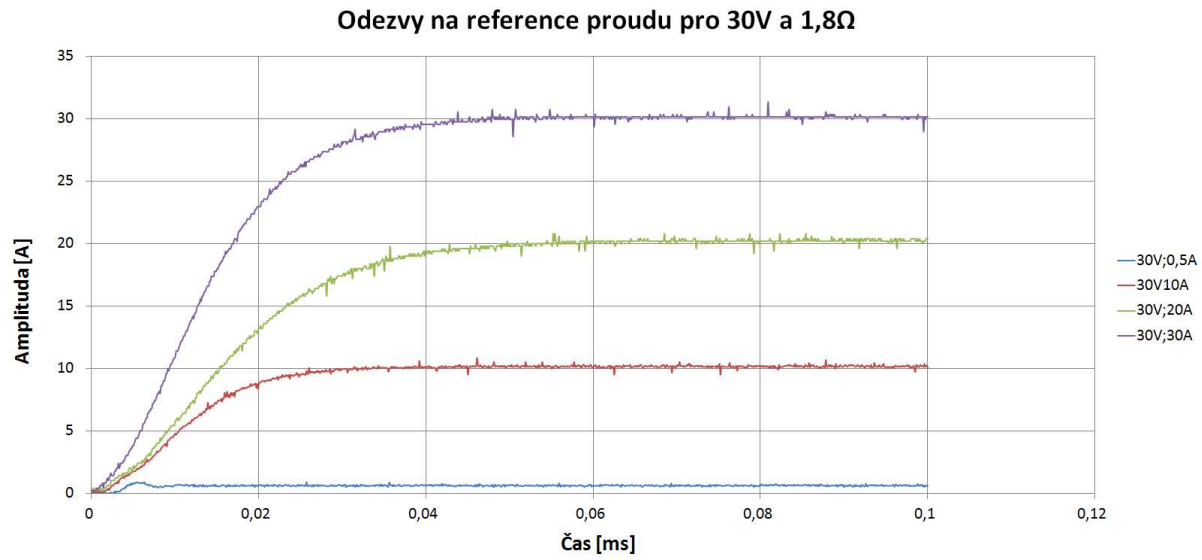
Obrázek 8.9: Reakce na reference pro 6V, 1,8Ω,Gain I=0,00100708, 1/U/R=-0,0996094



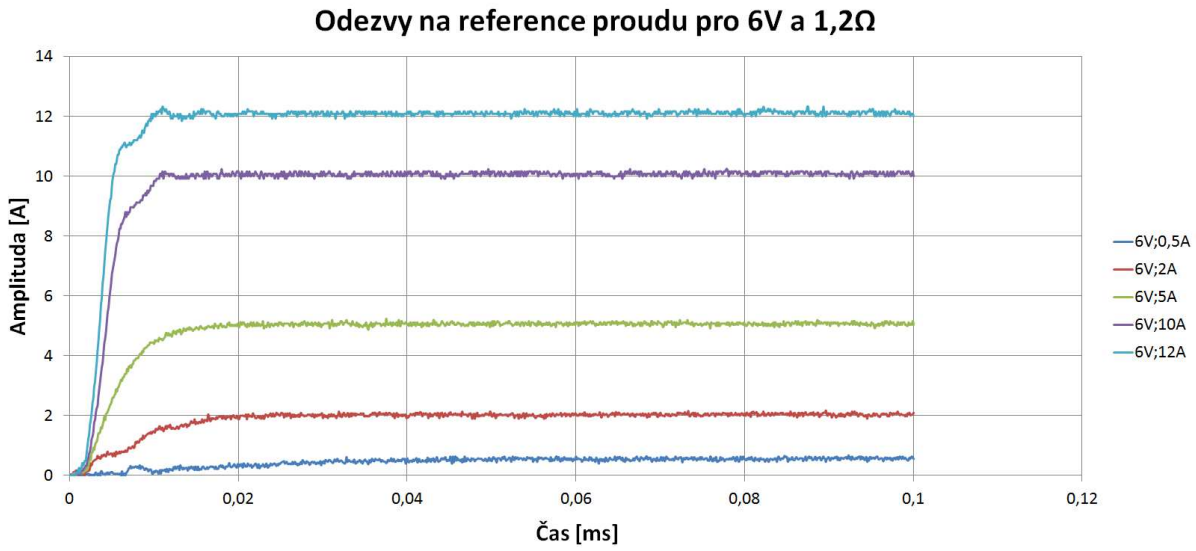
Obrázek 8.10: Reakce na reference pro 12V, 1,8Ω,Gain I=0,00100708, 1/U/R=-0,0507812



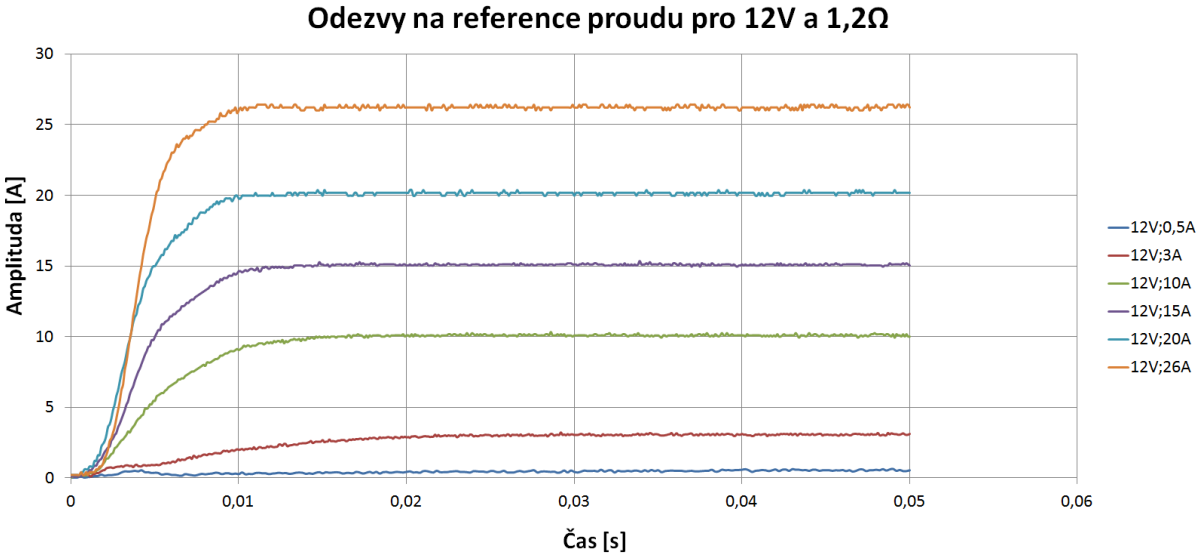
Obrázek 8.11: Reakce na reference pro 24V, 1,8Ω,Gain I=0,00100708, 1/U/R=-0,0253906



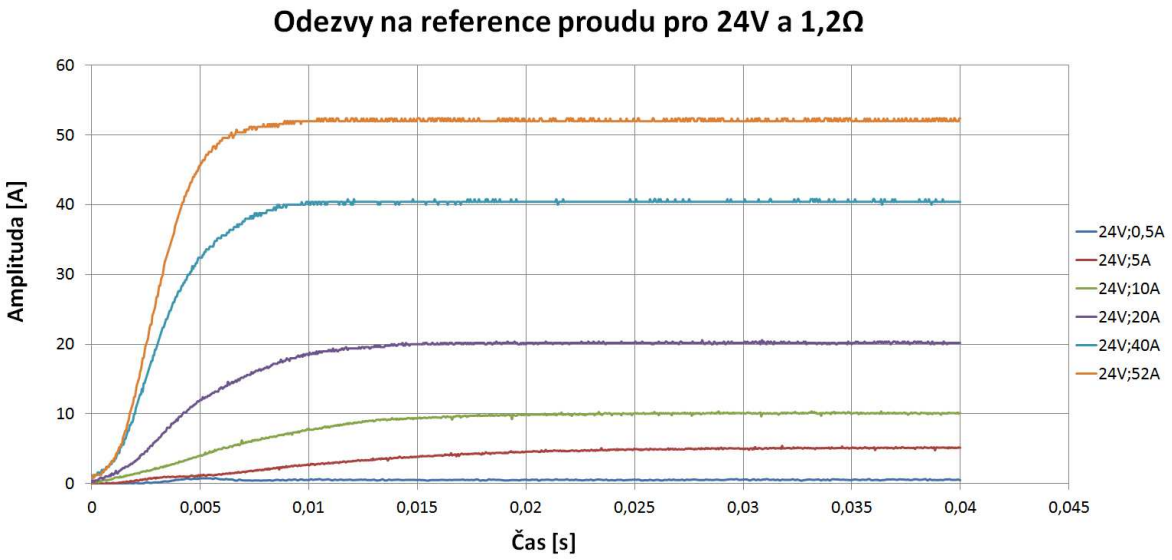
Obrázek 8.12: Reakce na reference pro 24V, 1,8Ω,Gain I=0,00100708, 1/U/R=-0,0195312



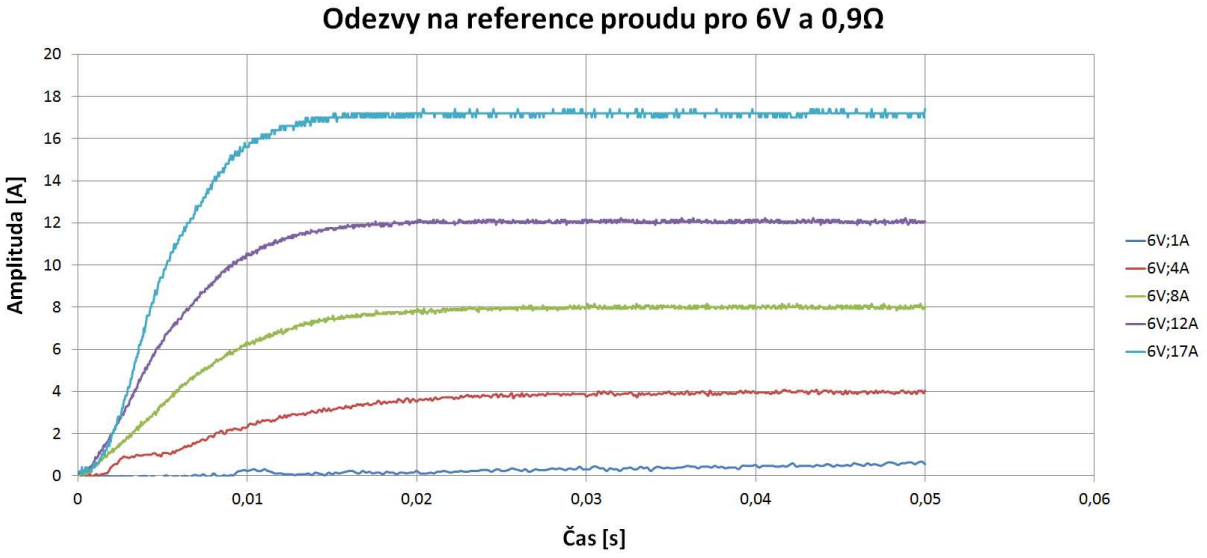
Obrázek 8.13: Reakce na reference pro 6V, 1,2Ω,Gain I=0,001464843, 1/U/R=-0,0664062



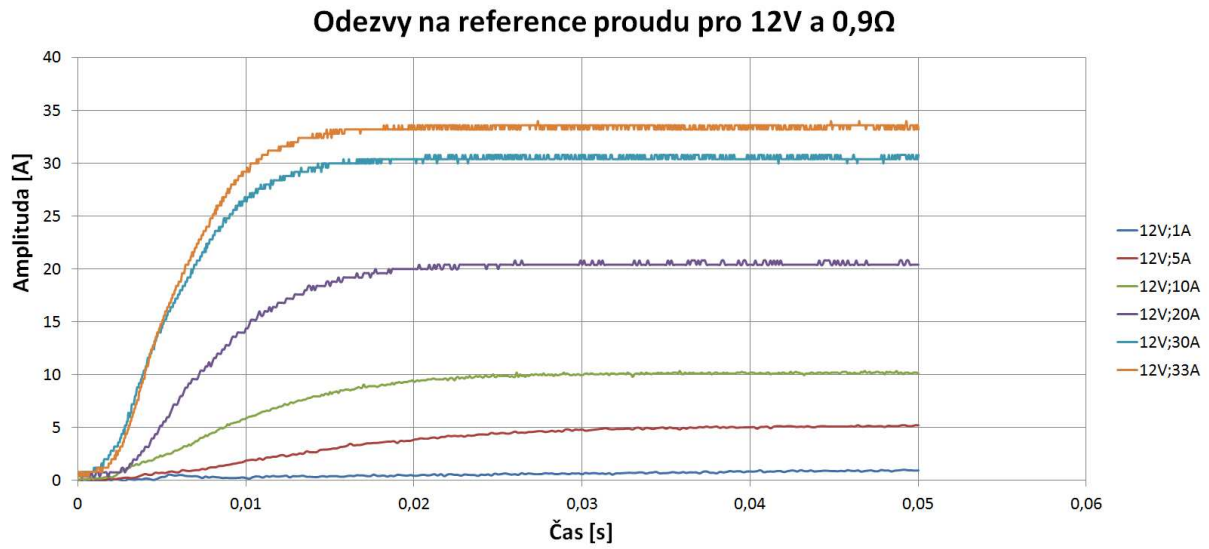
Obrázek 8.14: Reakce na reference pro 12V, 1,2Ω,Gain I=0,001464843, 1/U/R=-0,0332031



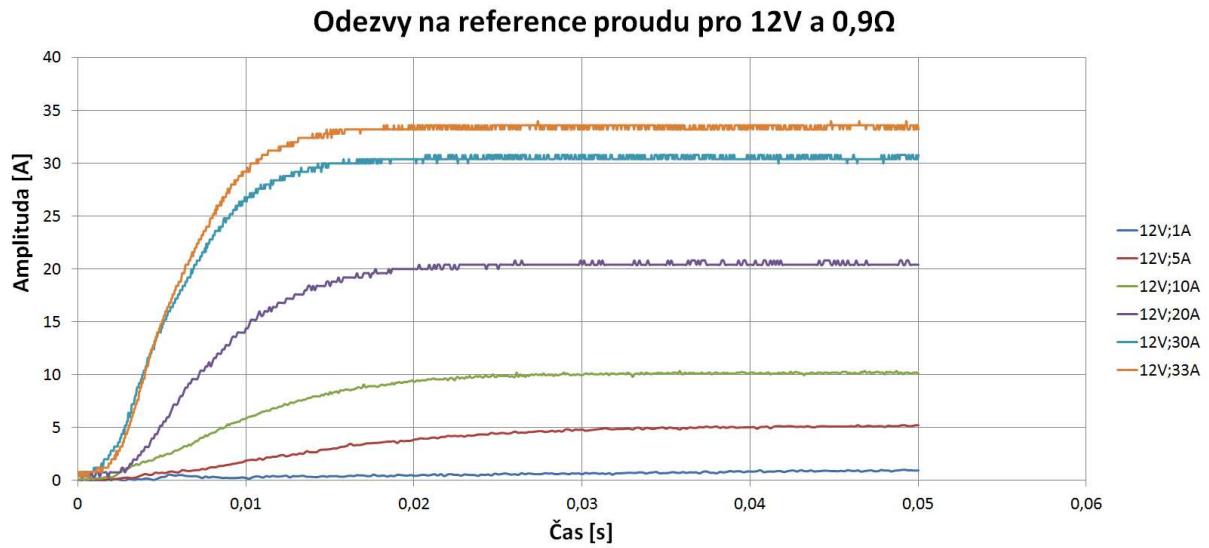
Obrázek 8.15: Reakce na reference pro 24V, 1,2Ω,Gain I=0,001464843, 1/U/R=-0,0166626



Obrázek 8.16: Reakce na reference pro 6V, 0,2Ω,Gain I=0,000976562, 1/U/R=-0,0507812



Obrázek 8.17: Reakce na reference pro 12V, 0,9Ω,Gain I=0,000976562, 1/U/R=-0,0253906



Obrázek 8.18: Reakce na reference pro 18V, 0,9Ω,Gain I=0,000976562, 1/U/R=-0,0175781

Z grafů vychází, že pro jednotlivé nastavení odporů je pro všechna napětí I Gain stejné, pouze 1/U/R se mění dle aktuálního napětí a při změně odporu se i I Gain mění. Z těchto měření jsem přišel na následující závislosti regulátoru:

Tabulka 8.4: Parametr regulátoru pro jednotlivé odpory na řízené zátěži

Odpor	Nastavené parametry
1,8 Ω	1/U/R= -1/U/1,8; I=0,00100708
1,2 Ω	1/U/R= -1/U/1,2; I=0,001464843
0,9 Ω	1/U/R= -1/U/0,9; I=0,000976562

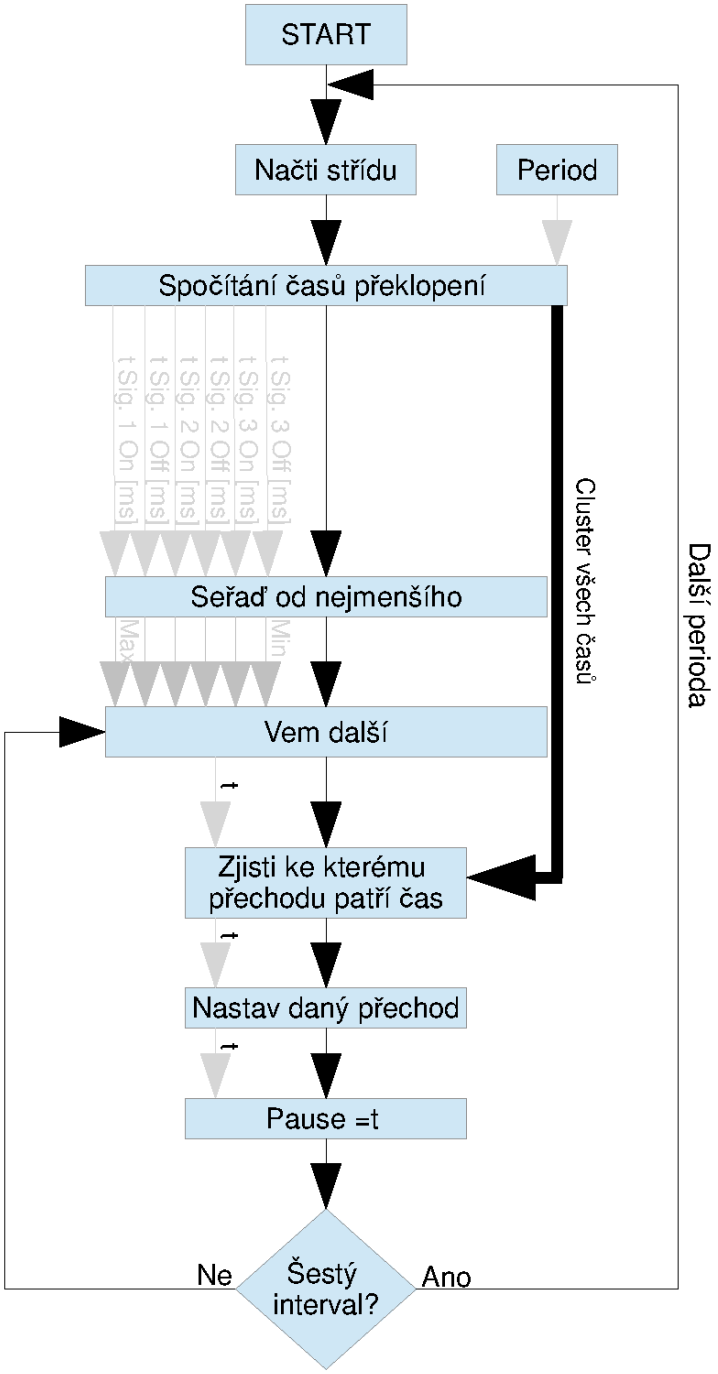
Z grafů jde dále poznat, že ustálení hodnoty probíhá do 50 ms, což je pro naše podmínky dostačující.

8.3.3 Přepínání regulátoru

Přepínání regulátoru probíhá dle nastavení v programu. Uživatel určí, jaký odpor je aktuálně navolen na řízené zátěži, a dle toho se nastaví hodnoty na I regulátoru. Za I regulátorem je generátor PWM signálu, který je připojen na jednotlivé sekce měniče.

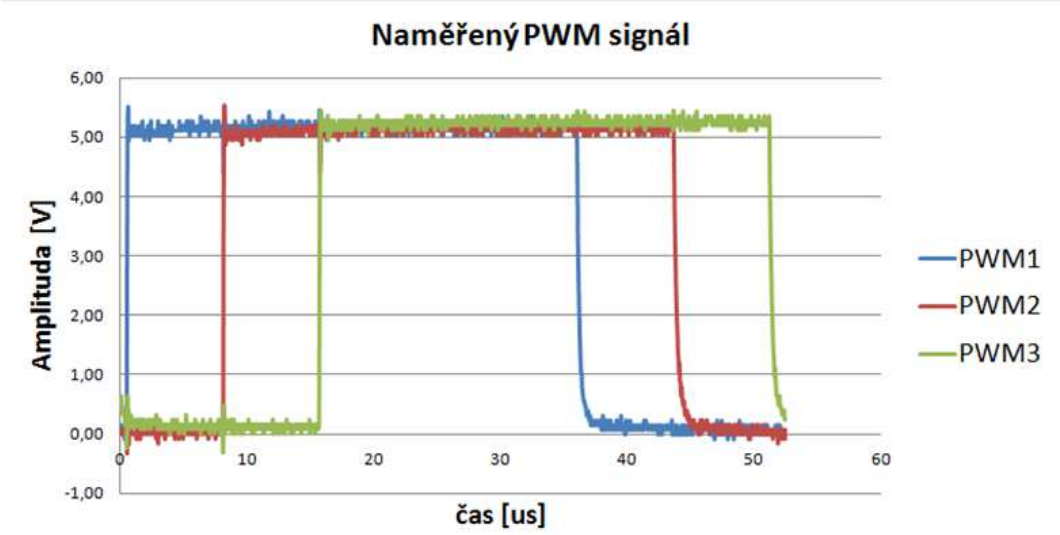
8.3.4 Řídicí signál PWM

Pro řízení zátěže pro vybíjení je využíván PWM signál. Jelikož se řízená zátěž skládá ze 3 sekcí, využíváme k řízení 3 PWM signály. Abychom eliminovali zvlnění proudu, je vhodné PWM signály rozprostřít po celé jejich periodě. Proto jsem se rozhodl použít 3 PWM signály, které budou mít vždy stejnou střídu, ale budou posunuty o jednu třetinu periody. Délka periody je stanovena na takovou hodnotu, aby byla mimo slyšitelné kmitočty a zároveň abychom ji byli schopni s daným hardwarem regulovat s dostatečnou citlivostí. Zvolená perioda byla nakonec stanovena na 50 kHz. Ke generování takového signálu jsem chtěl použít integrovaný generátor signálu přímo v LabView, ovšem tento generátor při dosažení maximální či minimální střídy bylo nutné restartovat. To pro mé účely bylo nevyhovující. Proto bylo nutné vyvinout vlastní algoritmus pro generování takových signálů. Vyvinutý algoritmus je založen na tom, že se na začátku každé periody vezme požadovaná střída a délka periody a spočítají se jednotlivé časy, kdy se signály lámou mezi 0 a jejich amplitudou. Tím vznikne 6 časů - pro každý signál vždy náběh a sestup v dané periodě.



Obrázek 8.19: Vývojový diagram generování PWM

Příklad výpočtu:
Frekvence: f = 50kHz (T= 20μs) Střída: D = 70%
Pro první signál je vždy začátek 0, a tedy konec spočítáme jako D·T , 0,7· 20=14μs.
Druhý signál se "roztahuje"na prostředku periody rovnoměrně vlevo i vpravo od středu. Střed periody je 10μs a délka pulsu je 14μs, tedy začátek pulsu bude $10-7=3\mu$s a konec 10+7=17μs.
Třetí signál se "roztahuje"od konce periody. Konec periody je na 20μs, a tedy začátek pulsu bude $20-14=6\mu$s a konec pulsu bude na konci periody 20μs.

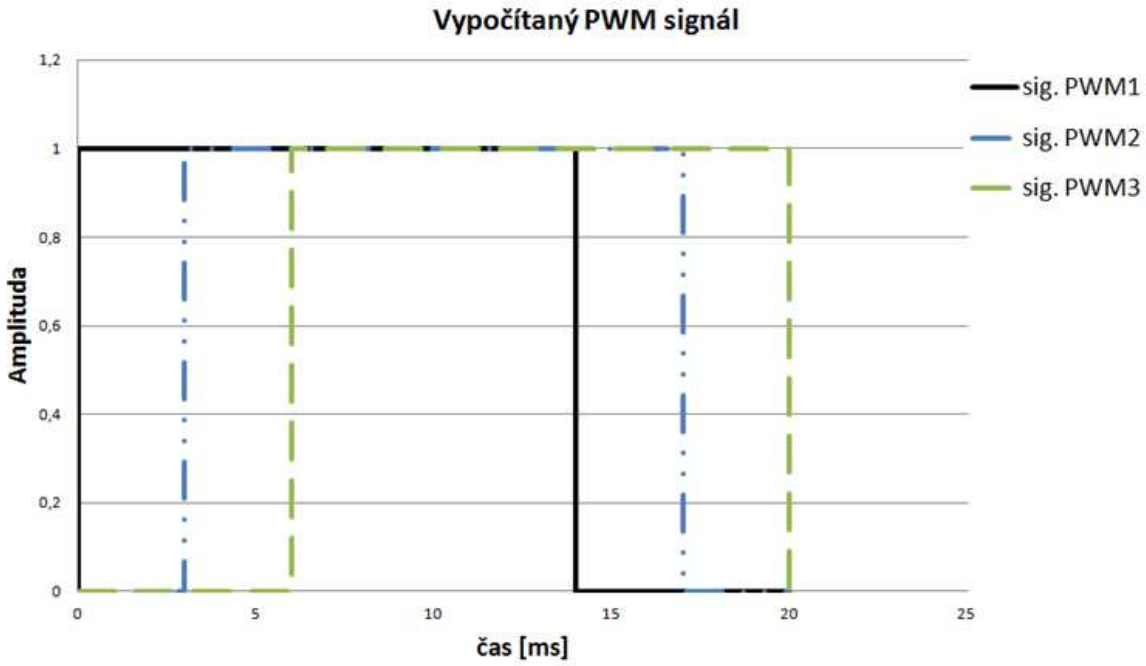


Obrázek 8.21: Naměřený průběh PWM

V algoritmu, který je vytvořen v LabView, je na začátku každé periody vyhodnoceno, zda střída signálu je 0 nebo 1, a pokud je, celý proces generování se zastaví a na výstupy se nastaví 0 nebo 1. Je to dáno tím, že marginální hodnoty způsobují v algoritmu překmity, a tak je lepší si tyto hodnoty do algoritmu nepustit a ošetřit je před ním.

Tabulka 8.5: Časy přechodů jednotlivých signálů mezi 0 a jejich amplitudou

Signál	Čas přechodu
Signál 1 do 1	0 μs
Signál 1 do 0	14 μs
Signál 2 do 1	3 μs
Signál 2 do 0	17 μs
Signál 3 do 1	6 μs
Signál 3 do 0	20 μs



Obrázek 8.20: Vypočítaný průběh PWM

Kapitola 9

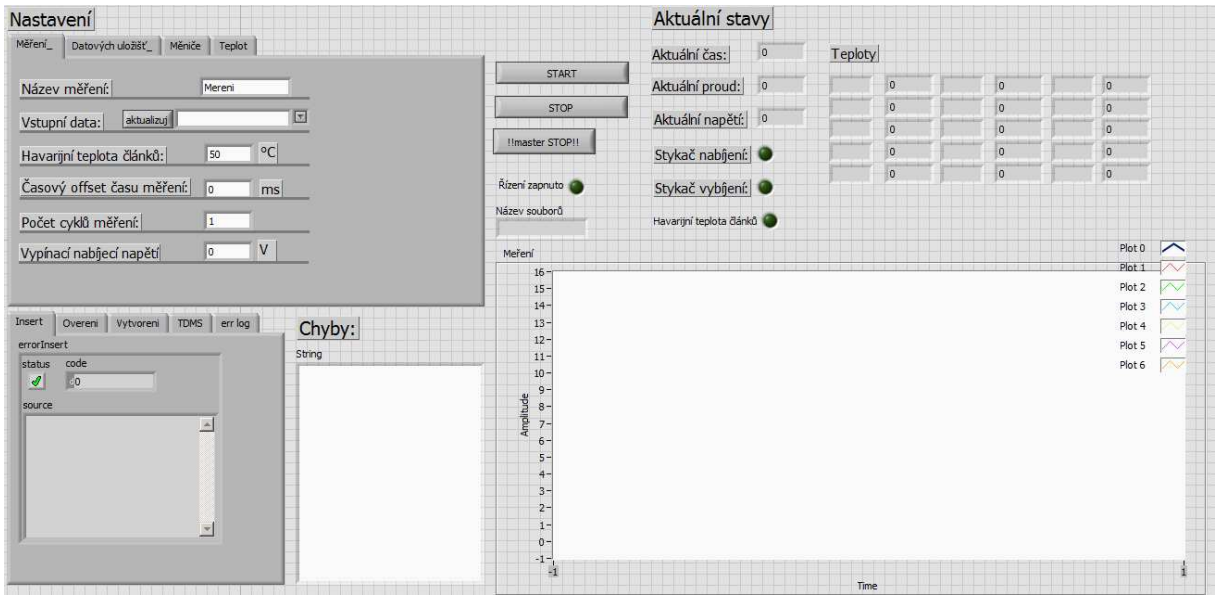
Popis práce s programem a ukázkové měření

9.1 Popis programu

Před testováním baterie je nutné si rozmyslet průběh požadovaného proudu a zanést ho do databáze tak, jak je popsáno v kapitole Datové úložiště. Po zanesení těchto dat do databáze je nutné si vyzkoušet komunikaci mezi cRiem a databázovým serverem. To je možné učinit tak, že v nastavení v sekci vstupní data stiskneme "aktualizuj". Pak se nám musí v combo boxu (vedle tlačítka "aktualizuj") zobrazit tabulky, které jsou v databázi validní pro vstupní data. Pakliže se neobjeví žádná tabulka, je problém buď v tom, že v databázi není validní tabulka pro vstupní data, nebo není možné se se serverem spojit. Pokud se spojení s databází nepodařilo, zobrazí se nám toto hlášení v menu vlevo dole pod záložkou ověření. V případě problému je namístě ověřit zadané parametry serveru v menu nastavení pod záložkou datová úložiště. Dále je vhodné vyzkoušet vzdáleně připojení k mysql serveru a vypsat si práva pro uživatele zadávaného do pole mysql serveru a pro IP adresu cRia. Po ověření výše zmíněného by mělo spojení fungovat. Identifikaci chyby můžeme také poznat z hlášení v menu ověření.

Popis jednotlivých položek v menu **Nastavení záložka Měření:**
Název měření – vyplníme název měření, který se nám promítne jako název tabulky (společně s časem začátku měření). Dále se pod tímto názvem vytvoří složka v paměti cRio, kam se ukládají dočasné soubory tak, jak je popsané v kapitole Datové úložiště.
Vstupní data – zde volíme z průběhů zadaných v mysql serveru. Pro načtení aktuálně validních tabulek je nutné stisknout tlačítko "aktualizuj".

Havarijní teplota článků – zadává se havarijní teplota článků. Tedy pokud překročí alespoň jedno teplotní čidlo umístěné ve spodní části rozvaděče u baterie tuto teplotu, přeruší se nabíjení a toto hlášení se napíše do textového pole string.
Časový offset času měření – časový offset od nuly, je využíván, pokud někdo nebo nějaká situace vypne započaté měření. Laborant v takové situaci najde čas odebrání posledního vzorku, podle kterého nastaví offset tak, aby měření pokračovalo od stejného bodu.
Počet cyklů – udává, kolikrát má cyklus proběhnout.
Záložka nastavení datových úložišť:
Zde je nutné vyplnit veškeré informace potřebné k autentizaci a přihlášení k sql serveru.
Záložka Nastavení měniče
Teplota zapínání ventilátorů– v průběhu měření dochází k vyhodnocování 3 teplot umístěných v měničích. Pakliže alespoň jedna z nich přesáhne teplotu nastavenou v této položce, dojde k zapnutí ventilátorů a k ochlazení měniče.
Zvolení aktuální konfigurace odporů - slouží pro přepnutí ke správnému PID regulátoru **Záložka Nastavení teplot**
Nastavení teplot souvisí s měřením teploty v rozvaděči v sekci baterie. Zde může být umístěno až 15 čidel. Po stisknutí na tlačítko aktualizace proběhne na této sběrnici 1wire vyhledání veškerých senzorů. Po vyhledání těchto senzorů se začne program zjišťovat teplotu jednotlivých senzorů, kterou vypisuje do GUI. Jelikož je pravděpodobné, že se vždy nebudou využívat veškerá teplotní čidla, je zde možnost zvolit pomocí Check boxů pouze logování té teploty, kterou potřebujeme. Ovšem pozor, havarijní teploty jsou vyhodnocovány pouze z čidel, která jsou vybrána pomocí check boxů. Každému čidlu můžeme také přiřadit jméno. To slouží pro identifikaci hodnot v záznamu databáze.
Další ovládací prvky:
Tlačítko start - započne měření. Pakliže stisknete toto tlačítko, všechny nastavovací tlačítka zešednou a nebude s nimi možno nijak manipulovat. Z ovládacích prvků zůstane aktivní tlačítko stop a master stop.
Tlačítko stop – ukončí měření.
Tlačítko master stop – ukončí veškeré regulační smyčky i smyčky pro zadávání hodnot a vypne program. Výstupní stavy se přenesou do defaultních hodnot, tzn. vše do nuly.



Obrázek 9.1: GUI

9.2 ukázkové měření

Před měřením bylo nutné nejdříve vytvořit vstupní hodnoty. Přihlásil jsem se k mysql serveru a vytvořil novou tabulku v databázi vstup_mereni

```
table mereni1 (cas int unsigned, proud double,
charge boolean, discharge boolean);
```

Následně jsem tuto tabulku naplnil požadovanými vstupními daty:

```
insert into mereni1 values
(0,0,true,false),
(0,3,false,1),
(30,3,false,1),
(60,3,false,1),
(120,4,false,1),
(180,0,true,false),
(190,0,true,false),
(220,0,true,false),
(250,0,true,false),
```

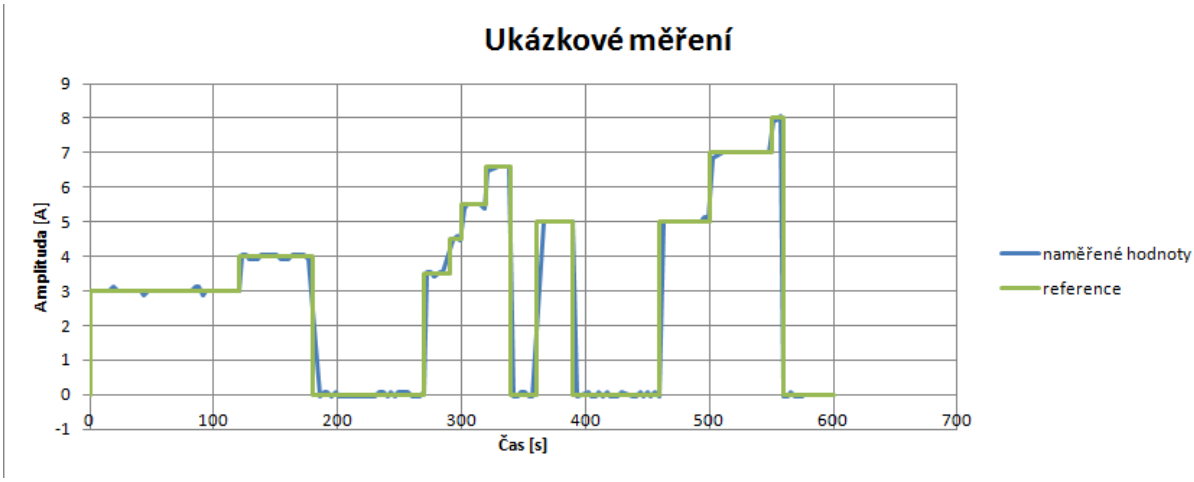
```
(270,3.5,false,true),
(290,4.5,false,true),
(300,5.5,false,true),
(320,6.6,false,true),
(340,0,true,false),
(360,5,false,true),
(390,0,true,false),
(410,0,true,false),
(420,0,true,false),
(460,5,false,true),
(500,7,false,true),
(550,8,false,true),
(560,0,true,false),
(580,9,false,true),
(600,0,true,false);
```

Dále jsem v programu u cRia stiskl u Měření-> vstupní data tlačítko aktualizovat a vybral jsem tabulku mereni1. Následně jsem vyplnil další hodnoty jako havarijní teploty apod. Zatrhl jsem teplotu, kterou jsem chtěl logovat. Konkrétně to byla teplota 1. Následně jsem spustil měření tlačítkem start. Po stisku tlačítka se rozsvítila indikace řízení zapnuto a v položce aktuální čas se začal načítat čas měření. V grafu se pomalu vykreslovaly aktuální průběhy. Po skončení měření zhasla indikace řízení zapnuto a na sql serveru bylo možno vyzvednout výstupní hodnoty. (výstup z měření je možné si také zkopírovat přímo z cRia v podobě tdms souborů pomocí ftp připojení).

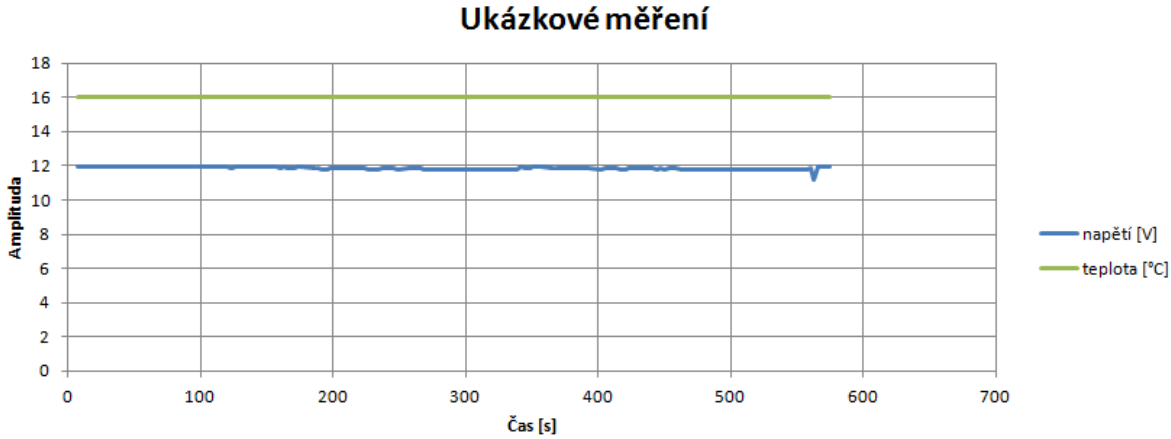
Pokud se připojíme k sql serveru pomocí ssh, můžeme exportovat hodnoty z databáze následujícím příkazem:

```
mysqldump -root -p vystup_mereni table_name > table_name.sql
```

a následně tento soubor zkopírovat na lokální počítač pomocí scp, u OS WINDOWS je možné použít nástroj winscp.



Obrázek 9.2: Srovnání reference proudu s realitou v ukázkovém měření



Obrázek 9.3: Průběhy monitorovaných veličin u ukázkového měření

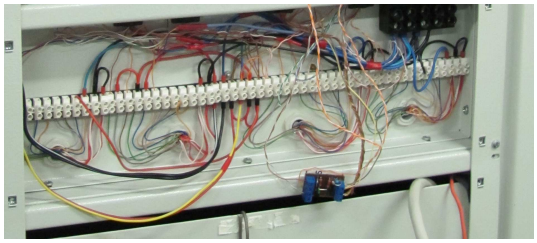
Odchylky od reference v grafu jsou dány vzorkováním, které je nastaveno na 3 sekundy. Pakliže bude měření dlouhé, vzorkovací frekvence bude zcela stačit pro monitorovaní dějů v bateriích.

Kapitola 10

Slaboproudé rozvody a výsledná podoba

10.1 Slaboproudé rozvody

Veškeré slaboproudé rozvody míří do malého prostoru zezadu rozvaděče. Zde jsou svorkovnice, do kterých jsou přivedeny veškeré IO z cRia. Ty jsou přivedeny do svorkovnice shora. Dále sem vedou dráty od LEM sond a 1wire sběrnice pro měření teploty baterie a je sem natažený 20žilový kabel od konektoru, který spojuje rozvaděč s řízenou zátěží.



Obrázek 10.1: Rozvaděč slaboproudu

Popis všech svorek v rozvaděči je uveden v příloze.

10.2 Rozvaděč zepředu

Celý rozvaděč je koncipován tak, aby bylo možné z něj jednotlivé prvky využít i mimo nabíjecí a vybíjecí stanici.

Obrázek 10.2: Přední panel vybíječe

Řízená zátěž má na sobě má oka a dvě svorky pro připojení zdroje. Dále zde je namontovaný konektor CANNON15, který je popsán v tabulce. Díky těmto výstupům přímo na řízené zátěži je možné využít řízenou zátěž i mimo tuto stanici tím, že připojíme jiný zdroj a můžeme připojit i jiné řízení. Pokud chceme připojit řízenou zátěž do systému stanice, stačí propojit cannon na stanici a cannon na řízené zátěži 1:1 kabelem a odpojit zdroj proudu od externích konektorů. Tak by vše mělo být opět ovládané vybíjecí a nabíjecí stanicí.

Tabulka 10.1: Tabulka zapojení konektoru CANNON

PIN	Funkce	Barva vodiče
1	+5V – napájení PWM	Hnědá od červené
2	GND PWM1	Šedá od červené
3	+5V – napájení PWM	Hnědá od bílé
4	GND-ventilátory	Bílá od oranžové
5	GND – 1WIRE	zelená od bílé
6	PWM1	Červená od šedé
7	PWM2	Modrá od bílé
8	PWM3	bílá od šedé
9	GND PWM2	bílá od modré
10	+5V – 1WIRE	zelená od červené
11	Ventilátor	oranžová od červené
12	GND – napájení PWM	Oranžová od bílé
13	GND - napájení PWM	Modrá od červené
14	1wire data	bílá od zelené
15	GND PWM3	Šedá od bílé

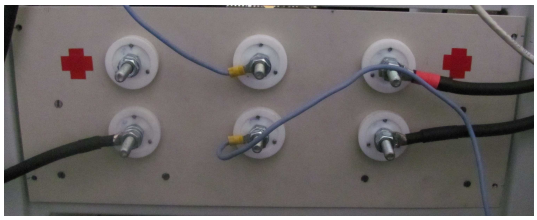
první barva je barva drátu a druhá barva drátu, který je sním v twist páru, tyto barvy korespodují s barvami vodičů v rozvodech rozvaděče

Dalším prvkem je nabíječ, který je přímo svou konstrukcí sestaven tak, aby se dal využít i mimo rozvody rozvaděče, stačí na svorky připojit jinou zátěž.

Obrázek 10.3: Přední panel nabíječe

Posledním prvkem jsou stykače. Tyto stykače mají také externí svorky na předním panelu. Prostřední svorky jsou od zdroje (baterie), levé svorky jsou od vybíječe a pravé

od nabíječe.



Obrázek 10.4: Přední panel stykačů

10.3 Dokumentace stavu

Obrázek 10.5: Fotografie před začátkem diplomové práce

Obrázek 10.6: Fotografie po skončení diplomové práci

Jediná část na rozvaděči, která není zatím zcela dokončena je silová část. Silová část je sice plně funkční, ale není finálně ukotvena v rozvaděči. Je potřeba dokoupit do rozvaděče DIN lišty a jejich uchycení a zabudovat tam zapojení silových prvků, které teď leží v rozvaděči na uchycené na provizorní DIN liště.

Kapitola 11

Závěr

Cílem diplomové práce bylo navržení testovacího zařízení pro testování velkých baterií. Navržená a realizovaná testovací stanice je založena na automatizovaném vybíjení a nabíjení akumulátorů dle předepsaných průběhů. Vzhledem k tomu, že stanice je navržena pro testování velkých baterií, musí tomu odpovídat i její parametry, kterými jsou maximální vybíjecí proud 100A a maximální nabíjecí proud 150A.
V existujícím rozvaděči jsem realizoval kabeláž jak silové části, tak pro komunikaci se senzory a pro řízení řízené zátěže.
Při volbě a návrhu senzorů pro tuto stanici jsem volil pro proud a napětí senzory takové, abych oddělil řídicí systém od baterie a zabránil tak možnému zničení řídicího systému. K monitoringu teploty jsem zvolil teplotní čídla připojené pro 1wire sběrnici a k těmto čidlům jsem naprogramoval vlastní vyhledávací algoritmus.
Pro řízení celé soustavy jsem zvolil řídicí systém cRio 9004. K řízení řízené zátěže jsem využil FPGA procesor integrovaný v cRiu. Do FPGA procesoru jsem naprogramoval generátor PWM signálu s posunem 1/3 periody, z důvodů snížení zvlnění zatěžovacího proudu. Dále kromě generátoru PWM signálu je v FPGA procesoru umístěn regulátor řízené zátěže realizovaný jako integrátor regulační odchylky. V FPGA procesoru je také umístěný kód pro komunikaci po 1wire sběrnici. Dalším úkolem řídicího systému je GUI, je možné k systému přistoupit jak vzdáleně, pomocí webového prohlížeče, tak pomocí nástroje od NI. Dále je v řídicím systému realizované zálohování a to jak do interní paměti řídicího systému, tak na externí sql server. Problém s délkou kompilace FPGA kódu jsem řešil migrací kompilace na výkonnější server mimo lokální síť.
Výsledkem práce je testovací stanice velkých baterií s řídicím softwarem, s dokumentací kabeláže v rozvaděči a sql serverem pro zálohování.
V další práci s touto stanicí by bylo dobré vytvořit databázi průběhů vybíjení a nabíjení

baterie pro testování baterií a také vytvořit automatické zpracování dat naměřených touto stanicí. Například identifikace modelů baterie na základě naměřených dat.

Literatura

[1] NAVE, Carl Rod. Lead-Acid Battery. *HyperPhysics* [online]. 2014 [cit. 2015-01-01]. Dostupné z: http://hyperphysics.phy-astr.gsu.edu/hbase/electric/leadacid.html

[2] BAJRACHARYA, Quree. *Dynamic modeling, monitoring and control of energy storage system.* Karlstad, 2013. Dostupné z: http://kau.diva-portal.org/smash/record.jsf?pid=diva2%3A608407&dswid=359. Diplomová práce. Karlstad University, Faculty of Technology and Science.

[3] NATIONAL INSTRUMENTS CORPORATION. Getting Started With LabVIEW FPGA. *National Instruments* [online]. 2014 [cit. 2015-01-01]. Dostupné z: http://www.ni.com/tutorial/14532/en/#toc1

[4] MALÝ, Martin. Sběrnice 1-Wire. In: *Hw.cz* [online]. 1997 - 2014 [cit. 2015-01-01]. Dostupné z: http://www.hw.cz/navrh-obvodu/rozhrani/sbernice-1-wiretm.html

[5] MALÝ, Martin. Algoritmus procházení sběrnice 1-Wire. In: Hw.cz [online]. © 1997 - 2014 [cit. 2015-01-01]. Dostupné z: http://www.hw.cz/teorie-a-praxe/dokumentace/algoritmus-prochazeni-sbernice-1-wire.html

[6] MAXIM INTEGRATED PRODUCTS, INC. 1-Wire Search Algorithm - Application Note - Maxim. Maxim Integrated [online]. 2014 [cit. 2015-01-01]. Dostupné z: http://www.maximintegrated.com/en/app-notes/index.mvp/id/187

[7] MARCOPOLO5. Native LabVIEW TCP/IP Connector for mySQL Database. In: NATIONAL INSTRUMENTS CORPORATION. National Instruments [online]. © 2014 [cit. 2015-01-02]. Dostupné z: https://decibel.ni.com/content/docs/DOC-10453

[8] SPECTRE_DAVE. 1-Wire. In: NATIONAL INSTRUMENTS CORPORATION. National Instruments [online]. © 2014 [cit. 2015-01-02]. Dostupné z: https://decibel.ni.com/content/docs/DOC-5572

[9] NATIONAL INSTRUMENTS CORPORATION. LabVIEW TM: Database Connectivity Toolkit User Manual [online]. 2008. [cit. 2015-01-01]. Dostupné z: http://www.ni.com/pdf/manuals/371525a.pdf

[10] VRLA battery [online], poslední aktualizace 13. prosince 2014 17:23 [cit. 1. 1.2015], *Wikipedie*. Dostupné z WWW: http://en.wikipedia.org/wiki/VRLA_battery

[11] Chan, H.L., "A new battery model for use with battery energy storage systems and electric vehicles power systems," *Power Engineering Society Winter Meeting, 2000. IEEE* , vol.1, no., pp.470,475 vol.1, 2000 doi: 10.1109/PESW.2000.850009

[12] J.F.Manwell, J.G.Mcgowan,EXTENSION OF THE KINETIC BATTERY MODEL OR WIND/HYBRID POWER SYSTEMS, *5th European wind energy association conference and exhibition, conference proceedings volume III*, 1994, [cit. 2015 1.1.] dostupné z: http://www.researchgate.net/publication/246153107_Extension_of_the_Kinetic_Battery_Model_for_WindHybrid_Power_Systems

[13] LOCKERBY, Patrick. Wilhelm Sinsteden - Inventor Of The Lead-Acid Battery. *Science 2.0* [online]. 2013 [cit. 2015-01-02]. Dostupné z: http://www.science20.com/chatter_box/wilhelm_sinsteden_inventor_leadacid_battery-112271

[14] GOODNIGHT, Jim. Understanding and Optimizing Battery Temperature Compensation. *SOLAR PRO* [online]. 2010 [cit. 2015-01-02]. Dostupné z: http://solarprofessional.com/articles/design-installation/understanding-and-optimizing-battery-temperature-compensation

[15] Lead-Acid Battery Resources. Lead-Acid Battery Info [online]. 2005 [cit. 2015-01-02]. Dostupné z: http://solarprofessional.com/articles/design-installation/understanding-and-optimizing-battery-temperature-compensation

[16] ITACA. Part 1: How Lead-Acid Batteries Work. A Guide To Lead-Acid Battries [online]. 2005 [cit. 2015-01-02]. Dostupné z: http://solarprofessional.com/articles/design-installation/understanding-and-optimizing-battery-temperature-compensation

[17] NATIONAL INSTRUMENTS CORPORATION. LabVIEW TM: Data Sheet CompactRIO Real-Time Embedded Controllers NI cRIO-9002, NI cRIO-9004 [online]. 2014. [cit. 2015-01-01]. Dostupné z:`http://www.ni.com/datasheet/pdf/en/ds-200`

[18] NATIONAL INSTRUMENTS CORPORATION. LabVIEW TM: Getting Started with CompactRIO - Offloading Signal Processing with LabVIEW FPGA [online]. 2011. [cit. 2015-01-01]. Dostupné z: `http://www.ni.com/white-paper/11200/en/`

[19] NATIONAL INSTRUMENTS CORPORATION. LabVIEW TM: NI LabVIEW FPGA Compilation Options [online]. 2011. [cit. 2015-01-01]. Dostupné z: `http://www.ni.com/white-paper/11573/en/`

[20] Jestem Grzesiek: Termometr do PC DS18B20 DS18S20 DS1820 [online]. 2011. [cit. 2015-01-01]. Dostupné z `http://grzesiek21.republika.pl/termo.htm`

[21] SPECTRE_DAVE. 1-Wire Interface for cRIO and sbRIO - DS18B20 or DS18S20. In: NATIONAL INSTRUMENTS CORPORATION. National Instruments [online]. © 2012 [cit. 2015-01-02]. ostupné z: `https://decibel.ni.com/content/docs/DOC-24136`

[22] Ptácek, V.: *Řízení systému pro měření ampérhodinové kapacity akumulátorových baterií.*ČVUT FEL, 2012.

[23] Ptáček, V.: *Řízení systému pro měření ampérhodinové kapacity akumulátorových baterií.*ČVUT FEL, 2012.

[24] Kmínek, M.: *Testovací pracoviště pro měření baterií - realizace.*ČVUT FEL, 2012.

[25] Eprona: *Uživatelský manuál, Modulový staniční napáječ HFM-A 80/100.*

Příloha A

Použitý software

NI Academic Site Licence Fall 2014 for Windows (univerzitní licence)- programování cRia
NI LabVIEW 2014 FPGA Module for Windows (univerzitní licence)- programování FPGA procesoru
ProfiCad (zdarma pro nekomerční účely) - kreslení schémat
MATLAB r2014 (licence katedry řídicí techniky) - modelování simulace, tvorba některých grafů
OpenOffice 4.1.1 (zdarma) - kreslení topologických schémat a vývojových diagramů
Microsoft Office Excel 2007 (soukromá licence)- kreslení některých grafů
ImageMagick-6.9.0-Q16 (freeware) - konverze obrázků
Texmaker 4.4.1 (freeware)- tvorba diplomové práce
MiKTeX 2.9 (freeware)- tvorba diplomové práce
InkScape (freeware)- tvorba některých obrázků
Oracle VirtualBox-4.3.20 (zdarma pro nekomerční účely) - k testování sql serveru, když jsem ještě neměl reálný server
mysql (freeware)
putty, winscp (freeware) - vzdálené připojení k linuxovému sql serveru

Příloha B

Obsah přiloženého CD

K této práci je přiloženo CD, na kterém jsou uloženy zdrojové kódy.

- zdrojove_kody_NI_labview
- data_z_mereni_osc
- Prilohy_v_el_podobe
- zdrojove_kody_matlab
- zdrojove_materialy_pouzite_v_textu_dp
- Software
- pouzite_zdrojove_kody
- literatura
- diplomova_prace.pdf

V každé složce je readme.txt, kde popisuji jak s obsahem složky zacházet a co je ve složce uloženo.

Příloha C

Seznam přiložených materiálů

- schéma rozvodů v rozvaděči
- popis svorkovnic v rozvodu slaboproudu