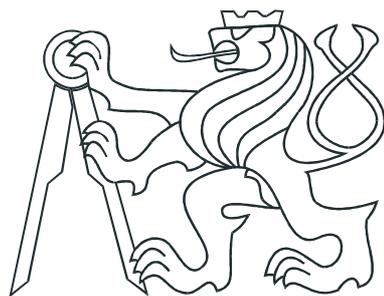


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ



BAKALÁŘSKÁ PRÁCE

Řízení systému pro měření ampérhodinové
kapacity akumulátorových baterií

Praha, 2012

Autor: Viktor Ptáček

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Viktor Ptáček**

Studijní program: Kybernetika a robotika

Obor: Systémy a řízení

Název tématu: **Řízení systému pro měření ampérhodinové kapacity akumulátorových baterií.**

Pokyny pro vypracování:

1. Prostudujte požadavky na automatické měřiče ampérhodinové kapacity akumulátorů.
2. Navrhněte vlastní systém řízení s využitím stávajícího HW (NI USB-6009)
3. Realizujte ukázkové měření s vámi navrženým systémem řízení.

Seznam odborné literatury:

[1] Normy související s měřením ampérhodinové kapacity baterií [2] Bakalářská práce: Matěj Kmínek, Testovací pracoviště pro nabíjení baterií.

Vedoucí: Ing. Pavel Hrzina, Ph.D.

Platnost zadání: do konce zimního semestru 2012/2013


prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 22. 12. 2011

Prohlášení

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

V Praze dne 25. 5. 2012


podpis

Poděkování

Děkuji především vedoucímu bakalářské práce Ing. Pavlu Hrzinovi Phd. za vedení a odbornou pomoc při vytváření této práce. A také bych chtěl poděkovat mé rodině za podporu v průběhu celého mého studia.

Abstrakt

Tato práce se zabývá automatizací již existujícího zařízení pro nabíjení a vybíjení akumulátorů. V této práci je navržen způsob regulace řízeného odporu a to pomocí procesoru STM8s. Dále je zde zpracována vizualizace, import a export dat pro nabíjecí a vybíjecí stanici, která již existuje na katedře. Program pro vizualizaci import a export dat je naprogramován v prostředí Labview.

Abstract

This bachelor project is dealing with designing control system for existing Battery Tester. The Battery Tester is a result of another bachelor project. The first part of this project is about controlling discharging element in battery tester. This element is controlled by microprocessor STM8s and uses PWM as a control signal. The second part of this bachelor project is about control of complete system for battery tester, for this control is development environment LabView used.

Obsah

Seznam obrázků	vii
Seznam tabulek	ix
1 Úvod	1
2 Popis problému, specifikace cíle	3
2.1 Popis problému	3
2.2 Specifikace cíle	3
3 Náhled na řešení	5
3.1 Schéma soustavy a jeho popis	5
3.2 Popis stávajících prvků testovací soustavy	6
3.2.1 Nabíječ	6
3.2.2 Měření proudu	6
3.2.3 Měření napětí	6
3.2.4 Měření teploty	7
3.2.5 Řídicí jednotka NI 6009	7
4 Řízená zátěž	9
4.1 Teorie	9
4.2 Modelování vybíječe	10
4.3 Model vs. realita	12
4.4 Další měření na vybíječi	15
4.5 Vybíječ a jeho tři sekce	16
4.6 Řízení vybíječe	16
4.6.1 Výběr vhodného mikrokontroléru	16
4.6.2 Způsob generování posunutých PWM signálů	17

4.6.3	Využití periferie mikrokontroléru a jejich popis	18
4.6.3.1	Časovač 1 TIM1	18
4.6.3.2	Časovač 2 TIM2	19
4.6.3.3	Problém s Discovery development kitem a výstupem TIM1	19
4.6.3.4	Problém se synchronizací	20
4.6.3.5	Analogově digitální převodník ADC1	20
4.6.3.6	Funkce programu	20
4.7	Interface pro řídicí obvod	22
5	Příprava a zpracování dat pro měření	25
5.1	Zadání hodnot uživatelem	25
5.1.1	Specifikace vytváření vstupních dat	26
5.1.2	Vytvoření formuláře a rozpoznání chyb zadaných uživatelem . . .	26
5.1.3	Ochrana před uživatelem	27
6	Program pro řízení celé soustavy	29
6.0.4	Něco málo o LabView	29
6.0.5	Představa o programu	29
6.0.6	Přední panel	30
6.0.7	Manuální a časový režim	31
6.0.8	Automatický režim	32
6.0.9	Program a jeho struktura	32
6.0.10	Měření napětí	34
6.0.11	Modifikovaný XY graf	35
6.0.12	Měření	36
7	Závěr	37
A	Obsah přiloženého DVD	I

Seznam obrázků

3.1	Schéma systému	5
3.2	Řídicí jednotka NI-6009	7
4.1	Náhradní schéma vybíječe	9
4.2	Fotografie vybíječe	11
4.3	Principiální schéma pro měření přechodové charakteristiky vybíječe	12
4.4	Přechodová charakteristika vybíječe $u_{bat}=12V$ PWM AI=3V	13
4.5	Přechodová charakteristika vybíječe $u_{bat}=12V$ PWM AI=3V -Modelica	13
4.6	Přechodová charakteristika vybíječe $u_{bat}=12V$ PWM AI=5V	14
4.7	Přechodová charakteristika vybíječe $u_{bat}=12V$ PWM AI=5V -Modelica	14
4.8	Převodní charakteristika	15
4.9	Graf průběhu čítače pro generování cent. PWM, převzato [8]	18
4.10	Graf průběhu čítače pro generování PWM, převzato [8]	19
4.11	Vývojový diagram - řízení PWM	21
4.12	Zapojení výstupu procesoru	22
4.13	Elektronické schéma interfacu	23
4.14	Navržená deska pološného spoje	24
5.1	Schéma cesty dat v procesu	25
5.2	Obrazovka uživatelského rozhraní	26
6.1	Přední panel	31
6.2	Blokové schéma programu	33
6.3	LabView diagram pro měření napětí	34
6.4	LabView diagram pro vlastní indikátor	35
6.5	graf srovnání předepsaného proudu s naměřeným	36

Seznam tabulek

3.1	Parametr přístroje Agilent 34410A	6
5.1	Význam hodnot v uživatelském rozhraní	27

Kapitola 1

Úvod

Cílem této práce je navržení řízení stávajícího zařízení pro testování moderních elektrických baterií. Toto zařízení bylo vyrobeno na základě předchozích bakalářských prací. Zařízení zatím není automatizované a měření na něm vyžaduje přítomnost laboranta. Touto prací by se výše uvedené mělo změnit. Laborant by měl jen specifikovat průběh nabíjení a vybíjení akumulátoru, respektive počet cyklů, které se mají vykonat. A následně by měla stanice a její řídicí systém obstarat vše potřebné, včetně bezpečnostních zásahů ošetření poruch....

Je nutné vytvořit program pro vizualizaci, export a import dat, k tomuto bude využit stávající modul NI 6009, který se v zařízení již nachází. Tento modul je z dílny National Instruments, tudíž jeho parametrizace a programování bude probíhat ve vývojovém prostředí LabView.

Dalším prvkem, který je nutné zprovoznit je řízená zátěž. Tato zátěž je již navržena a sestrojena, ale chybí k ní zařízení, které by ji ovládalo a bylo by schopné komunikovat s modulem NI 6009.

Kapitola 2

Popis problému, specifikace cíle

2.1 Popis problému

Na katedře elektrotechnologie se nachází zařízení pro testování baterii. Toto zařízení ve stávajícím stavu je ovládáno zcela laborantem a není zde významnější podíl automatizace. V rámci této bakalářské práce by mělo dojít k plnému automatizování tohoto zařízení tak, aby na laborantovi byl jen počáteční import dat o požadovaném průběhu nabíjení a konečný export dat a následné zpracování těchto dat do protokolu o testu baterii.

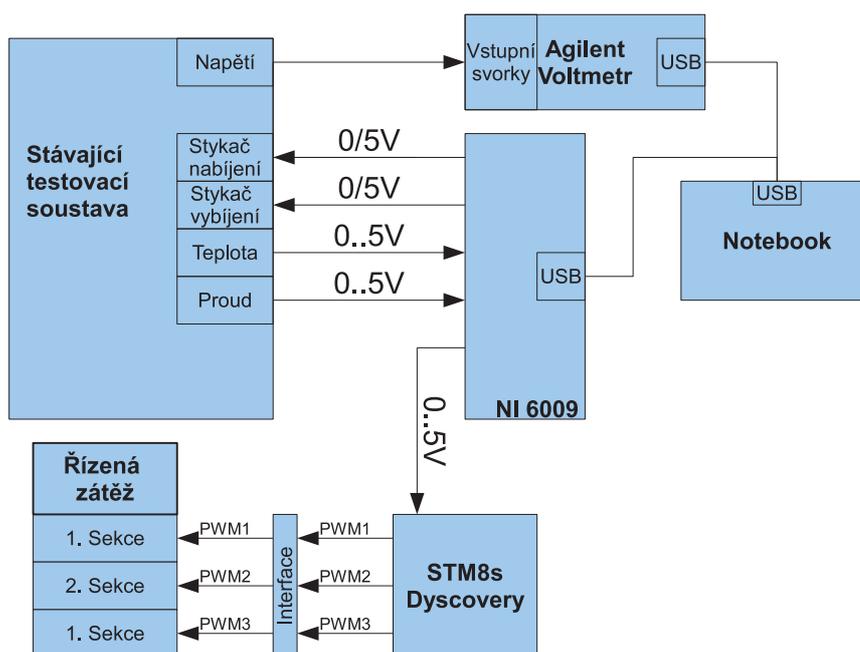
2.2 Specifikace cíle

Cílem této práce je vytvořit programové vybavení k již existující měřící soustavě. Navrhnout řízení v programovacím jazyku LabView. a dále navrhnout a realizovat řešení řízené zátěže. Přípravek řízené zátěže je již pro tuto soustavu navržen, ale chybí mu procesor pro řízení jeho chodu. Řízení zátěže bude zajišťovat procesor, který bude generovat požadované PWM signály do jednotlivých částí vybíječe (soustavy řízených odporů).

Kapitola 3

Náhled na řešení

3.1 Schéma soustavy a jeho popis



Obrázek 3.1: Schéma systému

Stávající systém je navržen tak, aby jej bylo možno automaticky ovládat. Hlavními ovládacími prvky tohoto systému jsou dva stykače. Sepnutím jednoho, nebo druhého se měřený akumulátor připojí buď na nabíječ, nebo k vybíjecímu obvodu. Dále stávající testovací soustava dovoluje měřit proud a teplotu

3.2 Popis stávajících prvků testovací soustavy

3.2.1 Nabíječ

V systému je zabudovaný nabíječ HFM-A 80/100 od firmy Eprona a.s [13]. Tento nabíječ má maximální výstupní stejnosměrné napětí 80V a může dodávat až 100A. Tento nabíječ nemá žádný interface, kterým by mohl komunikovat s okolím, tudíž pro nabíjení je nutné, aby obsluha zařízení zadala požadované hodnoty tak, aby vyhovovaly testované baterii. Program pak jen může nabíječ buď stykačem připojit, nebo odpojit.

3.2.2 Měření proudu

Pro měření proudu je použita proudová sonda LEM HASS 500-S [9]. Rozsah sondy je do 900A. V rozvaděči je touto sondou protažen kabel 3x, takže její výsledný proud je nutný nakonec vydělit 3. Toto protažení je provedeno proto, aby se procházející proud posunul blíže hodnotě nominálního proudu, který pro sondu definoval výrobce. Následující vztah určuje převod mezi měřeným napětím a proudem:

$$I = (U_{out} - 2,5) \cdot \frac{800}{3} - 0,7$$

3.2.3 Měření napětí

K měření napětí je zde využit multimetr Agilent 34410A od firmy HP [11]. Výhodou tohoto přístroje je jeho USB interface. Proto je možné multimetr připojit k počítači. Navíc k němu firma National Instruments dodává ovladače pro vývojové prostředí LabView, díky nimž se dá tento přístroj plně ovládat z počítače.

Parametr	Hodnota
Měřicí rozsah pro stejnosměrné napětí	100mV ... 1000V
Chyba měření pro stejnosměrné napětí	0,0035%

Tabulka 3.1: Parametr přístroje Agilent 34410A

3.2.4 Měření teploty

K soustavě jsou připojena dvě teplotní čidla LM235 [12]. Zapojení v čidle si lze představit jako dvě spojené Zenerovy diody. Jak je známo, Zenerovo napětí je závislé na teplotě. Po přiložení napětí je možné pozorovat mezi diodami změny napětí, které odpovídají změnám teploty.

Výstupem tohoto senzoru je napětí $10\text{mV}/^\circ\text{K}$. Jelikož v systému máme pro analogové hodnoty minimální rozsah -1 až 1V a maximální -10 až 10V , bylo nutné připojit přípravek, který toto napětí přizpůsobí. Byl k tomu použit již vyrobený prvek, který v sobě má dva operační zesilovače zapojené tak, aby výstupní napětí bylo 0V až 10V a 0.1V odpovídala 1°C .

3.2.5 Řídicí jednotka NI 6009

Celé zařízení je kompletně řízené z řídicí jednotky NI-6009 [10] od National Instruments. Jednotka je plně programovatelná. Je vhodné ji parametrizovat v prostředí LabView nebo LabWindow. Komunikaci s okolím zajišťuje 8 analogových vstupů, u kterých je možné různě nastavit rozsahy (minimální rozsah v rozmezí -1V až 1V , maximální rozsah v rozmezí -10V až 10V). Dále je možné využít dva analogové výstupy s pevným rozsahem 0 až 5V . Pro dvoustavové ovládání zde je 12 konfigurovatelných výstupů. Každý z nich může být nastaven buď jako výstup, nebo jako vstup.

Do této jednotky míří veškeré měřené signály našeho systému s výjimkou napětí. To měří přístroj Agilent 34410A, který přímo komunikuje s počítačem.



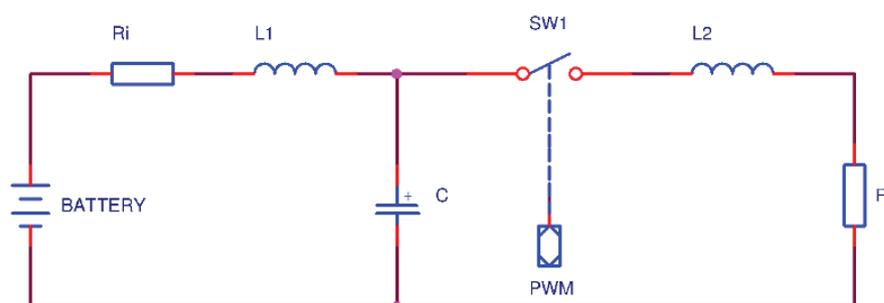
Obrázek 3.2: Řídicí jednotka NI-6009

Kapitola 4

Řízená zátěž

4.1 Teorie

Jako řízená zátěž je v sestavě využit vybíječ, který již byl na katedře elektro technologie sestrojen v rámci předchozích diplomových prací.



Obrázek 4.1: Náhradní schéma vybíječe

Jak jde vidět ze schématu přímo na baterii je připojen LC filtr. Kondenzátor C zde znázorňuje kondenzátorovou baterii, která je složená z 21 kondenzátorů s kapacitou $470\mu\text{F}$. Součástí filtru je také cívka. Dále je zde naznačen odpor, tento odpor fyzicky v soustavě není, ale reprezentuje nám vnitřní odpor baterie a odpor vinutí cívky. Pokud by zde tento odpor nebyl, došlo by k rozkmitání LC obvodu. Dále je ve schématu naznačen elektronický spínač, který slouží k regulování zátěže. Velikost zátěže určuje střída signálu PWM, nebo-li jak dlouho je spínač sepnutý, nebo vypnutý. Další a to poslední část obvodu reprezentuje zatěžovací odpor. Tento odpor je reprezentován odporem a indukčností. Jeho indukčnost je poměrně velká, jelikož konstrukčně je tento odpor vyroben

jako navinutý odporový drát.

Díky LC filtru dochází k tomu, že proud zátěží není konstantní a jeho průběh závisí na střídě PWM signálu, ale i přesto je proud odebíraný baterii konstantní, pokud je konstantní střída a perioda PWM signálu.

4.2 Modelování vybíječe

K modelování vybíječe jsem využil jeho náhradní schéma. Popsal jsem jej následujícími rovnicemi:

$$\begin{aligned} u_{bat} &= Ri \cdot i_{bat} + u_c + L_1 \frac{di_{bat}}{dt} \\ i_{bat} &= C \cdot \frac{u_c}{dt} + PWM \cdot i_{L2} \\ u_C \cdot PWM &= L_2 \frac{di_{L2}}{dt} + R \cdot I_{L2} \end{aligned}$$

vysvětlení proměnných použitých v rovnicích: u_{bat} - napětí baterií

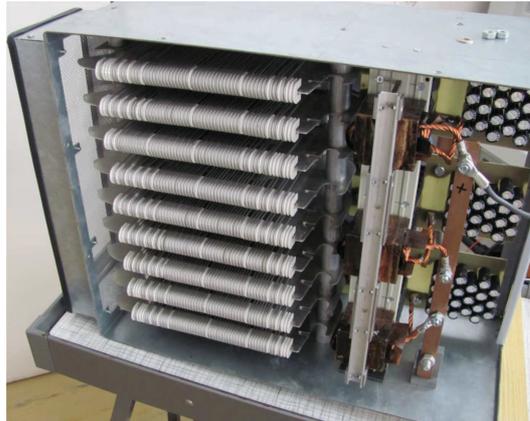
i_{bat} - proud baterie

u_c - napětí na kondenzátoru

i_{L2} - proud cívkou L2 nebo-li proud odporem R

PWM - je signál PWM, který má hodnotu buď 1 nebo 0.

Pokud se podíváme na soustavu diferenciálních rovnic, zjistíme, že nejsou lineární. Nelinearitě nám tam zanáší signál PWM. Jelikož celý systém se bude provozovat na celé škále střídě tohoto signálu, není možné provést linearizaci a přepsat tento systém do stavového popisu. Proto jsem se rozhodl jej přenechat v tomto tvaru a dále jej simulovat jako nelineární systém. Pro simulaci tohoto systému jsem použil program OpenModelica.



Obrázek 4.2: Fotografie vybíječe

Zdrojový kód pro modelicu:

```

model vybijec
  parameter Real L1 = 3.1e-005, C = 21 * 0.00047, L2 = 9.1e-005;
  parameter Real R = 1.8, Ri = 0.09, u1 = 12;
  Real PWM, iL2, uc, ibat;
  Real uPWM=3;
  Modelica.Blocks.Sources.Pulse pulse1(width = 100, period = 10,
  startTime = 0.2);
  Modelica.Blocks.Sources.SawTooth pulse2(amplitude = 4.999999,
  period = 1 / 20000, startTime = 0.2);

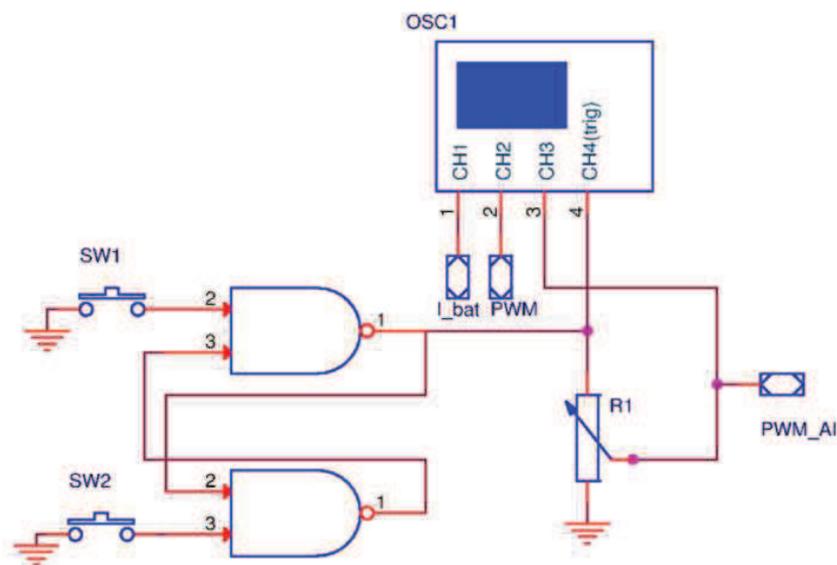
equation
  PWM = if pulse2.y - pulse1.y*uPWM <= 0 then 0 else 1;
  u1 = L1 * der(ibat) + uc + Ri * ibat;
  ibat = C * der(uc) + PWM * iL2;
  uc * PWM = L2 * der(iL2) + R * iL2;
end vybijec;
  
```

Parametry ve zdrojovém kódu jsou analogické ke schématu. Navíc ve zdrojovém kódu jsou i parametry systému. Prvním z parametrů je L1, což je indukčnost cívky L1. Tuto indukčnost jsem dostal v popisu vybíječe. Hodnota odporu R je daná nastavením vybíječe, lépe řečeno jeho sekce. Model systému je dělán pro odpor 1,8Ω. Dalším parametrem je odpor Ri, který jsem odhadl na 0.09Ω, dalším parametrem je u1, neboli napětí baterie,

v mém modelu jsem uvažoval napětí 12 V, jelikož 12voltové baterie se budou v soustavě nejčastěji testovat.

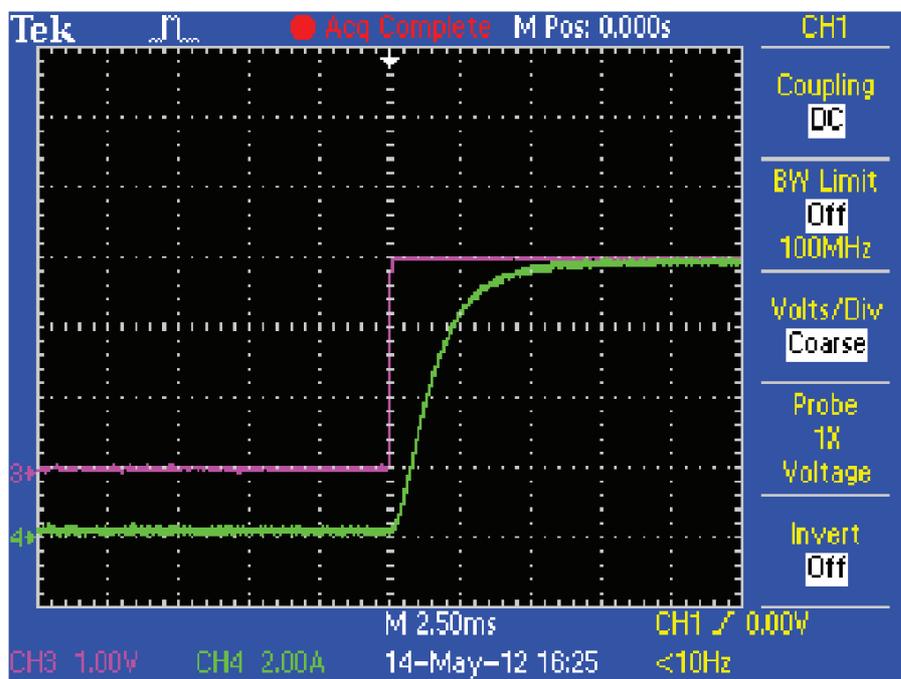
4.3 Model vs. realita

Správnost modelu jsem testoval na přechodové charakteristice systému. Pro tuto přechodovou charakteristiku jsem vytvořil schéma s elektronickým spínačem tak, abych mohl synchronizovat osciloskop se startem PWM signálu. K tomuto jsem využil RS klopný obvod.

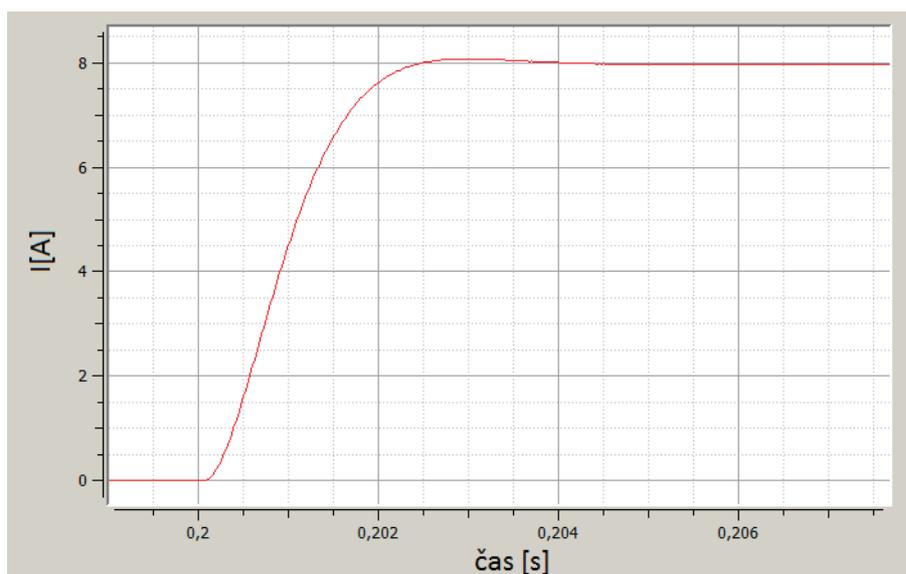


Obrázek 4.3: Principiální schéma pro měření přechodové charakteristiky vybíječe

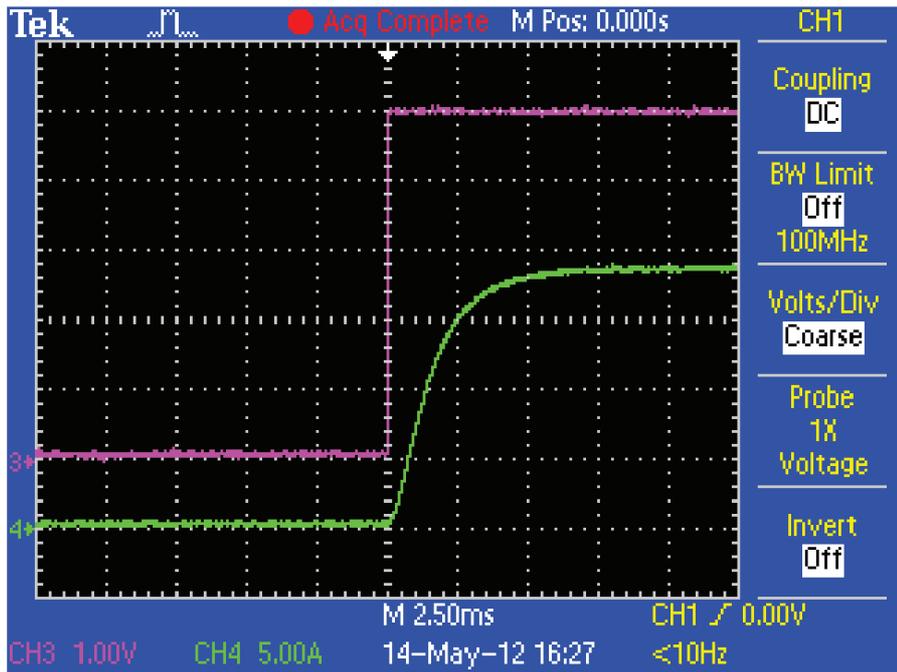
Zde jsou pro srovnání průběhy z Modelicy a průběhy naměřené na reálném systému:



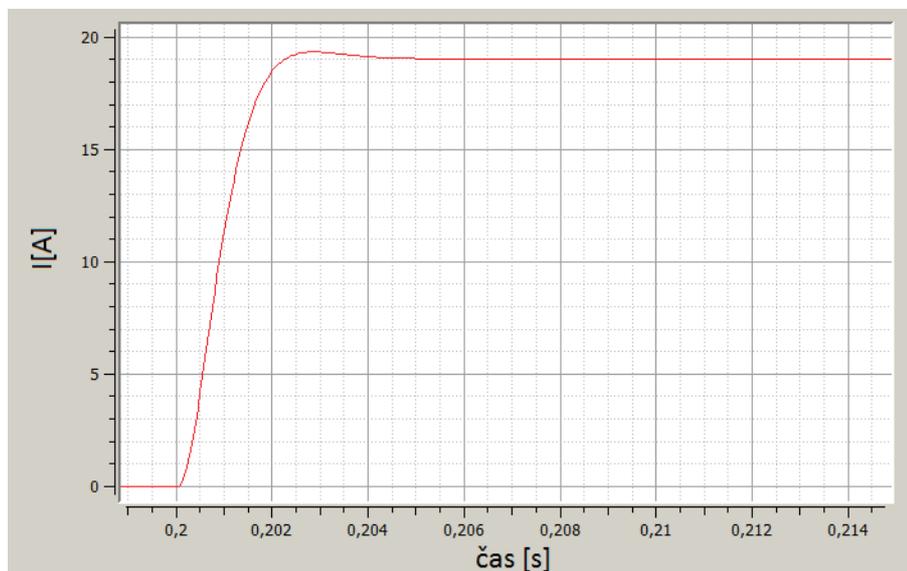
Obrázek 4.4: Přechodová charakteristika vybíječe $u_{bat}=12V$ PWM
AI=3V



Obrázek 4.5: Přechodová charakteristika vybíječe $u_{bat}=12V$ PWM
AI=3V -Modelica



Obrázek 4.6: Přechodová charakteristika vybíječe $u_{bat}=12V$ PWM
 $A_I=5V$



Obrázek 4.7: Přechodová charakteristika vybíječe $u_{bat}=12V$ PWM
 $A_I=5V$ -Modelica

Z průběhů lze poznat, že jejich ustálená hodnota je takřka totožná. A doba ustálení

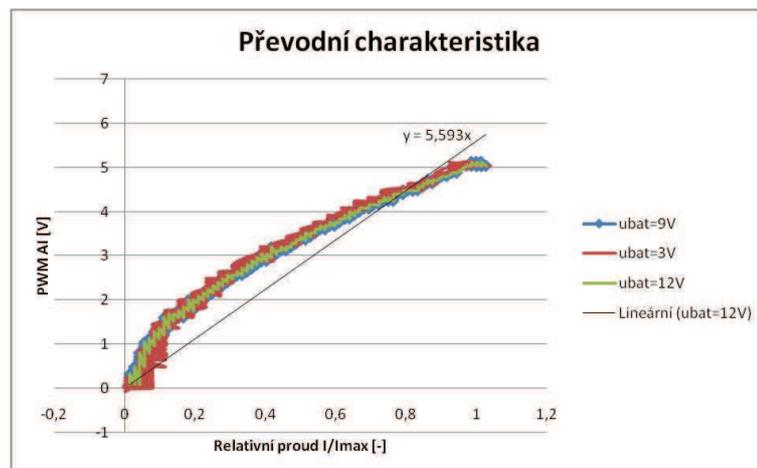
je také na všech průbězích přibližně stejná a to 5ms. Tím bych zhodnotil model jako dostatečnou aproximaci reálného systému.

4.4 Další měření na vybíječi

Další měření na vybíječi jsem prováděl za účelem zjistit převodní charakteristiku systému. Tedy kolik mám dát napětí do PWM AI, abych dostal určitý proud. To jsem zjišťoval tak, že jsem měřil průběh PWM AI a proud zároveň a poté vynesl hodnoty do grafu. Mnou použitý osciloskop neměl mód X-Y, proto jsem musel nastavit sequenční měření, které jsem nastavil asi na 10s. Po tuto dobu jsem potenciometrem měnil PWM AI a osciloskop zaznamenával průběh jak PWM AI, tak i procházejícího proudu. Tyto hodnoty jsem poté vynesl do grafu tak, že na ose x byl proud a na ose y napětí PWM AI. Zdá se, že je to naopak. Přeci měření bylo takové, že jsem měnil PWM AI a měřil proud. Ovšem z grafu potřebuji dostat rovnici, do které dosadím proud a budu mít napětí, které musím přivést na PWM AI, abych dostal požadovaný proud.

Toto měření jsem udělal pro různá napětí. Proto jsem všechna měření znormalizoval, tedy sloupec s proudem každého měření jsem vydělil maximálním proudem sloupce. Tím jsem dostal v každém sloupci pro proud rozsah 0-1. Poté jsem veškeré hodnoty vynesl do grafu. Pomocí Excelu jsem našel spojnicí trendu, po zkoušení logaritmického průběhu a proložení polynomem jsem zjistil, že nejlepší bude průběhy proložit lineárně.

Překrytí všech tří průběhů značí, že se systém chová stejně pro různá napětí baterie.



Obrázek 4.8: Převodní charakteristika

4.5 Vybíječ a jeho tři sekce

Náš vybíječ je složen ze 3 sekcí, což má dvě výhody. Udělat jednu sekci, aby zvládla proudy 100A, není zas tak lehké, navíc jsme zde limitováni maximálními hodnotami součástek... Proto je lepší udělat více méně výkonných sekcí a zapojit je paralelně a tím dostat dostatečný výkon. Další výhodou je, že každá sekce má svůj vlastní ovládací tranzistor, takže se dá ovládat nezávisle na těch ostatních. Takhle možnost ovládání nese jednu výhodu - výkon, který je spotřebováván, můžeme rozložit do periody relativně rovnoměrně a tím se "přiblížit" součtem proudů na rezistorech stejnosměrnému proudu. Je totiž lepší, než aby se všechnen výkon "spálil" hned na začátku periody, a pak celý obvod čekal na další začátek periody. Mohlo by se stát, že by se tímto proud odebíraný z baterie začal zvlňovat. Proto je výhodnější, aby se v průběhu periody proud odebíral rovnoměrně. V našem řízení nemůžeme dosáhnout plně rovnoměrného odebírání proudu (měřeného před tranzistorem), přece jen řízení je pomocí PWM signálu. Ale tomuto "rovnoměrnějšímu" odběru proudu můžeme napomoci tak, že PWM signály jdoucí do jednotlivých sekcí rozdělíme v periodě tak, aby se nepřekrývaly, pokud je to možné.

4.6 Řízení vybíječe

Z výše uvedeného by bylo nejlepší, aby se signály ke spínacím prvkům k jednotlivým sekcím vybíječe nepřekrývaly. Proto by měly být signály fázově posunuty o $1/3$ periody. Ke spínání je využit PWM signál, který generuje mikropočítač.

4.6.1 Výběr vhodného mikrokontroléru

Nejdříve jsem musel vytyčit požadované parametry, které musí mít řídicí procesor, aby zvládl to, co potřebuji. Ke komunikaci s vnějším světem je potřeba, aby procesor měl analogový vstup. Tento analogový vstup bude propojen s řídicím modulem NI 6009 a bude připojen na jeho analogový výstup. Tento výstup má výstupní parametry: 0 až 5V, s typickým výstupním proudem 5mA a zkratovým proudem 50mA [10]. Jelikož odběr analogového vstupu u procesorů je v jednotkách uA, není v tomto problém. Po analogovém vstupu je potřeba, aby daný procesor měl možnost hardwarově generovat PWM signál. Takže řídicí procesor musí mít 3 časovače, které budou schopny generovat PWM

signál. Nejdříve jsem se zaměřil na procesory, které jsou k dispozici na katedře a bylo by možné je co nejjednodušším způsobem použít. Při hledání jsem narazil na Development kit discovery-STM8s. Jeho parametry byly ideální. Měl analogový vstup, dále měl tři časovače, které dokázaly generovat PWM signály. Navíc tento Development kit se prodává za cenu kolem 200,- Kč, takže by se dal použít i pro nasazení v aplikaci. K tomuto je také tento kit přizpůsoben. Vlastně obsahuje dva obvody. Jedním z nich je programátor založený na obvodu STM32 a druhým je samotný STM8[8] s dodanými periferiemi.

4.6.2 Způsob generování posunutých PWM signálů

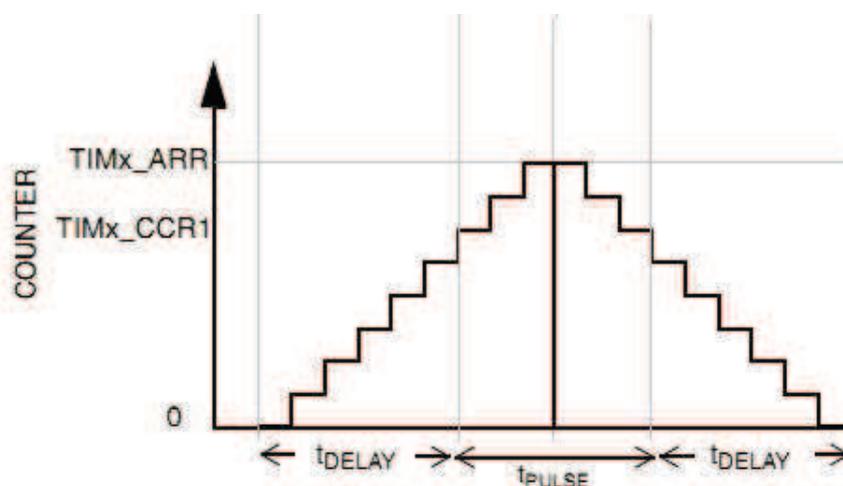
Byly zde dva možné způsoby, jakými se dají generovat takové posunuté PWM signály, které byly potřeba. Prvním z nich je pustit první PWM signál a na základě jeho přerušení synchronizovat čítače dalších dvou PWM signálů. Offsety čítačů by se nastavily podle změřených hodnot tak, aby PWM signály byly posunuté dle vytyčených podmínek. Druhý způsob, který jsem použil, je postavený na základě vlastnosti jednoho z časovačů, který je integrovaný v tomto procesoru. Jedná se o funkci PWM center align. Tato funkce umožňuje generovat PWM signál, který je zarovnán na střed periody. Čili první PWM signál se bude "roztahovat" na středu periody. Druhý bude narůstat od začátku periody a třetí bude narůstat od konce periody k jejímu začátku. Tím dosáhnu požadovaného efektu, aby se pwm signály překrývaly jen velice málo.

4.6.3 Využití periferie mikrokontroléru a jejich popis

Před započítím programování jsem se seznámil s veškerými parametry daného procesoru, k tomu jsem využil datasheety [7][8]. Ze kterých v této části čerpám.

4.6.3.1 Časovač 1 TIM1

Tento časovač je nejvybavenějším časovačem na procesoru. Díky tomu, že jeho čítač je schopen čítat i dolů, je to takzvaný updown counter, je možné generovat pwm signál, který je vycentrovaný na prostředku periody.

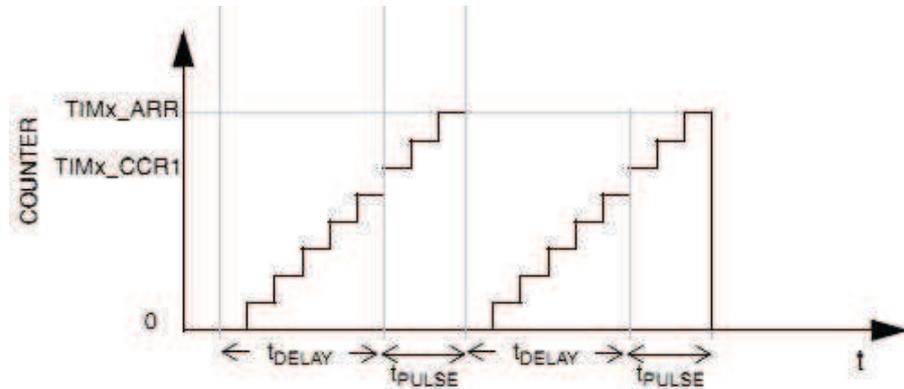


Obrázek 4.9: Graf průběhu čítače pro generování cent. PWM, převzato [8]

Na obrázku je vidět, že se timer časuje podle čítače, který je taktován podle hodinového signálu. Tento hodinový signál je nastavitelný. Při začátku čítání začíná na nule a čítá nahoru. Po dosažení hodnoty TIM1_ARR se čítač přepne do modu down a začne čítat dolů (perioda signálu se tedy rovná $2 \cdot \text{delka hodinového signálu} \cdot \text{TIM1_ARR}$). Výstupní signál je generován porovnáním s hodnotou v registru TIM1CCR1. Pokud je hodnota v registru větší než hodnota čítače, pak je signál v log. 0, pokud je hodnota v registru TIM1_CCR1 menší než hodnota čítače, pak je výstup v log.1. Ovšem takovéto generování má jednu nevýhodu. Pokud nastavíme velikost pulzu na 1 pak výstupní pulz bude trvat dva takty čítače a ne jen jeden. Tenhle timer je v naší aplikaci nastavený do módu COUNTERMODE_CENTERALIGNED2, a jeho kanál 1, který používáme je nastaven jako PWM2, což nám umožňuje, abychom měli výstup jaký požadujeme.

4.6.3.2 Časovač 2 TIM2

Tento časovač může čítat jenom jedním směrem a to nahoru.



Obrázek 4.10: Graf průběhu čítače pro generování PWM, převzato [8]

Na obrázku je vidět, jak časovač pracuje. Tento časovač jsem využil ke generování dvou signálů. U každého timeru je totiž možné využít několik kanálů. Jednotlivým kanálům lze nastavit určité vlastnosti. Všechny kanály mají stejnou periodu, ale je možné pro každý kanál nastavit, zda běží v módu PWM1 nebo PWM2. Jeden mód určuje, jak dlouho je na začátku periody log. 1 a druhý mód určuje, jak dlouho od začátku periody bude logická nula. Pokud máme nastavenou periodu na 800 pulzů, je potřeba nastavit délku pulzu pro PWM1 na požadovanou délku a u PWM2 na 800 mínus požadovaná délka pulzu. Pak nám teprve vzniknou 2 signály, kdy jeden se bude roztahovat od začátku periody a druhý od jejího konce. Z popisu TIM1 a TIM2 je zřejmé, že TIM1 musí mít 2x více pulzů než TIM2. Je to dané tím, že potřebujeme dojít k první komparaci a pak i ke druhé.

4.6.3.3 Problém s Discovery development kitem a výstupem TIM1

Při programování a rozcházení PWM signálu generovaného pomocí TIM1 jsem vše měl nastaveno správně, ale stále jsem na výstupu procesoru nemohl změřit ten signál, který jsem potřeboval. Byla tam jen konstantní hodnota napětí. Po prostudování materiálů dodaných k development kitu jsem zjistil, že zrovna na těchto výstupech je připojen dotykový senzor. To se dalo jednoduchým zásahem do desky změnit. Jednalo se o odstranění jednoho odporu a o přepojení dvou drátových propojek.

4.6.3.4 Problém se synchronizací

Po parametrizaci těchto výstupů a jejich zobrazení na osciloskopu jsem zjistil, že bych potřeboval tyto signály synchronizovat. Signály generované pomocí TIM2 byly synchronizované, ale rozjízďedly se právě k signálu generovaného TIM1. Když jsem si procházel datasheet, zjistil jsem, že hardwarová synchronizace časovačů v tomto mikrokontroléru chybí. Lépe řečeno u této řady mikrokontrolérů je hardwarová synchronizace mezi timery TIM1, TIM5 a TIM6. Ovšem náš mikropočítač je osazen jen TIM1. Tudíž jsem musel k synchronizaci využít přerušení jednoho z PWM signálů. Jelikož se chystám při tomto přerušení nastavit counter jednoho PWM signálu, musím k vyvolání tohoto přerušení použít ten, který je 2x pomalejší než ten druhý. Tudíž přerušení vyvolává PWM signál generovaný pomocí časovače TIM1. U nastavení čítače TIM2 je nutné zvolit určitý offset. Tento offset jsem určil z měření na osciloskopu. Metodou zvyšování a snižování tohoto offsetu jsem umístil PWM signály přesně tam, kam jsem potřeboval.

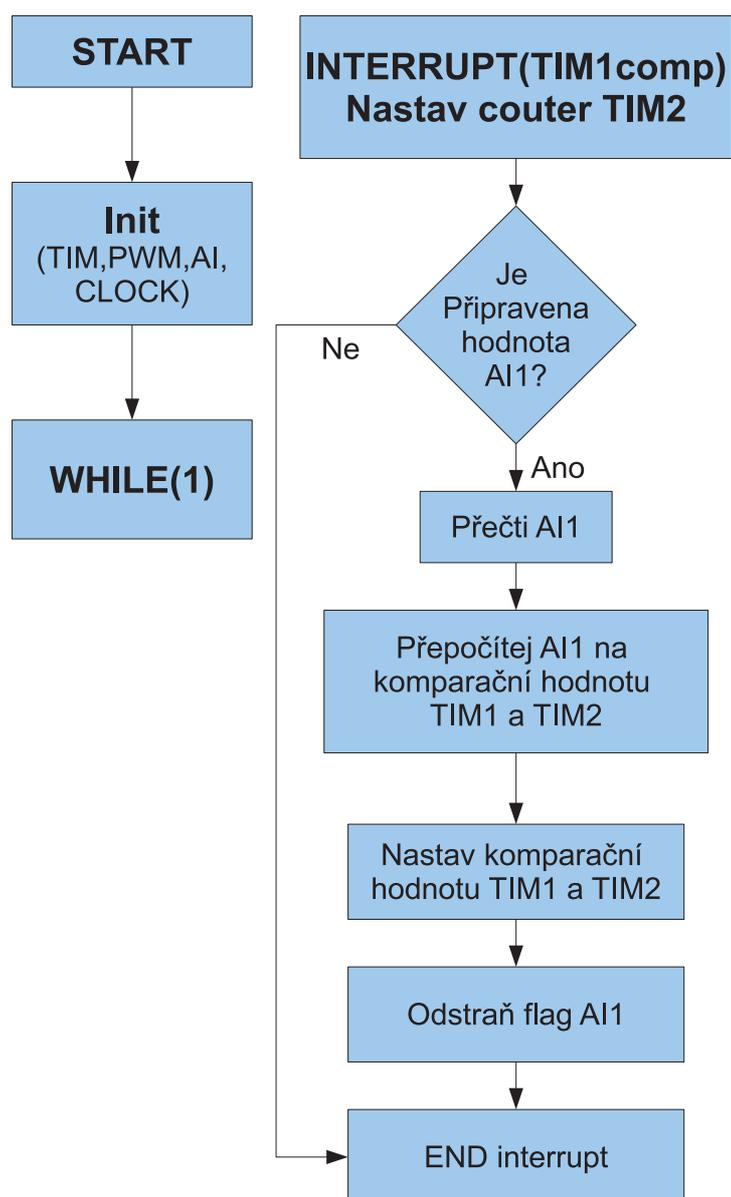
4.6.3.5 Analogově digitální převodník ADC1

Na mikrokontroleru je umístěn analogově digitální převodník, který v naší aplikaci využívám ke komunikaci s řídicí jednotkou NI 6009. Tento ADC převodník je 10-ti bitový a je možné jej provozovat ve dvou módech. První mód je single a druhý continues. Jelikož chceme měřit hodnotu v čase, nastavil jsem mód continues. Dále je zde možné zvolit buffrovaný mód nebo nebuffrovaný. Zvolil jsem mód nebuffrovaný, což znamená, že od vyvolání startu převodu převodník převádí hodnotu. Po převedení vyvolá přerušení a nastaví tzv. FLAG na bit EOC (end of conversion). Po restartování FLAGu se začne další převod. Jelikož je podmínkou našeho programu mít synchronizované PWM výstupy a právě tyto výstupy, lépe řečeno jejich střídání, chceme regulovat vstupem z řídicí jednotky je nutné synchronizovat i toto nahrávání hodnoty střídání. Takže zpracování analogové hodnoty z ADC převodníku neprovádím v přerušení vyvolaném ADC převodníkem, ale přerušením PWM1 signálu, který používám k synchronizaci PWM signálů. Abych zajistil to, že bude hodnota z ADC převodníku připravena, táži se před čtením této hodnoty, zda je EOC aktivní. Potom vezmu hodnotu z ADC převodníku a přenásobenou příslušnou konstantou ji nahraji do registrů, které mi určují střídání PWM signálů.

4.6.3.6 Funkce programu

Funkce programu je velice jednoduchá. V main funkci se nastaví potřebné periferie procesoru - timery a jejich výstupy se nakonfigurují jako PWM signály, nakonfigurují se

jejich periody... Dále se hodiny procesoru nastaví na nejvyšší možný kmitočet a nakonec proběhne nastavení analogového vstupu. Pak program vstoupí do nekonečné smyčky. Další chod obstarává přerušení. Používám jen jedno a to přerušení TIM1 při komparaci. Je to dáno tím, že perioda přerušení je násobkem periody ostatních časovačů.



Obrázek 4.11: Vývojový diagram - řízení PWM

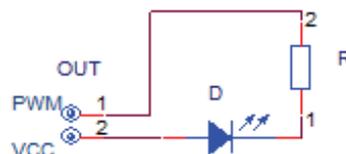
4.7 Interface pro řídicí obvod

Tento obvod zajišťuje přizpůsobení výstupů a vstupů řídicího procesoru STM8s. Dále jsem po oddělení programátoru a samotného procesoru s jeho periferiemi chtěl zachovat možnost dalšího přeprogramování a to bez nutnosti odpojení napájení. Jedním z úkolů Interfacu je zajištění napájení obvodu. Je zde ochrana proti přepólování realizovaná pomocí diody. Pak je zde zapojen vyhlazovací kondenzátor a stabilizátor. Stabilizátor jsem volil dostatečně výkonný, jelikož kromě procesoru budou napětím ze stabilizátoru napájeny i další prvky systému. K napájení dalších prvků systému je na plošném spoji vyvedena svorkovnice.

Pro development kit je na plošném spoji vytvořena pomocí dvouřadých zásuvkových lišt patice. Díky nesymetrii umístění těchto lišt nelze procesor dát naopak. Dalším úkolem desky je poskytnout procesoru požadované napájení. Procesor má mnoho vývodů, na které je nutné připojit napájecí napětí nebo GND. Ovšem na desce kitu jsou tyto piny spojeny. Proto je možné si vybrat kteroukoliv nožku pro napájení a pro zem a připojit tak jen dva piny "pouzdra".

Dalším úkolem desky je přizpůsobení a ochrana analogového vstupu pro procesor. Ochrana tohoto vstupu je zde realizovaná pomocí dvou Shottkyho diod, které zajišťují to, aby napětí na tomto vstupu nebylo větší než napájecí napětí, nebo menší než GND. Volil jsem Shottkyho diody, jelikož mají menší úbytek a jsou rychlé. Ochrana Shottkyho diod je zde provedena přes 560Ω odpory. Pokud dioda sepne bude na těchto odporech úbytek a tím bude diodou procházet menší proud. Dále je zde dle doporučení výrobce mezi vstup a zem připojen kondenzátor.

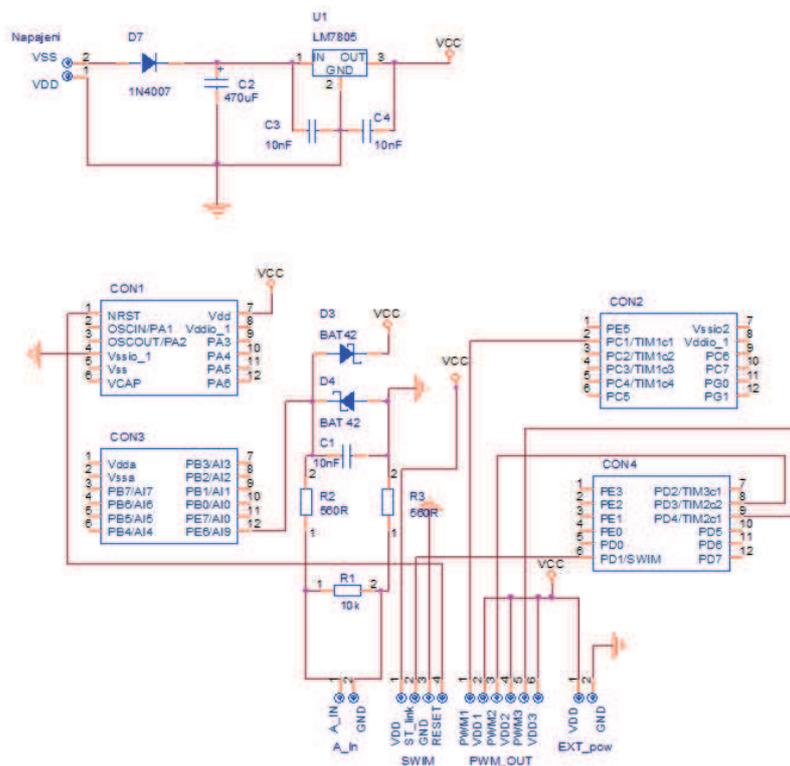
Na desce je realizované propojení výstupů PWM signálů se svorkovnicí. Tyto výstupy jsou HS, tudíž není nutné přizpůsobení. Tyto výstupy mají za úkol pouze spínat diodu v optočlenu vybíječe. Pro nižší proud výstupem procesoru je dioda zapojena takto:



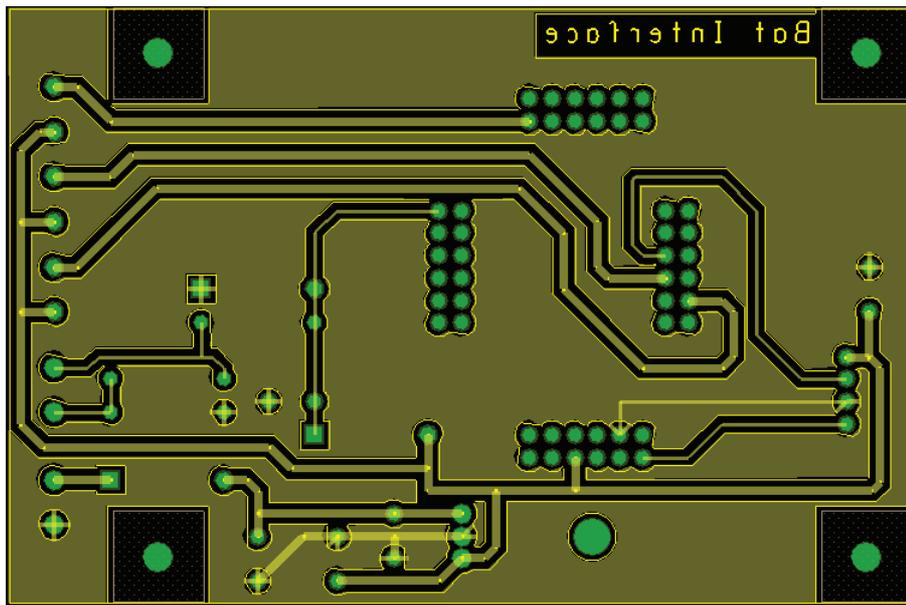
Obrázek 4.12: Zapojení výstupu procesoru

Z obrázku je zřejmé, že dioda je sepnutá pokud je PWM signál v nule.

Plošný spoj dále zajišťuje možnost přeprogramování a to programátorem, který byl součástí development kitu. Programátor procesoru neposkytuje napájení. Pro funkčnost musí naopak obvod procesoru poskytnout napájení programátoru a nejedná se tak o přímé spojení napájení desky procesoru s napájením USB sběrnice. Další výstupy nutné k programování SWIM, GND a NTRRS jsou přivedeny na konektor na okraji desky plošného spoje a je možné je tak jednoduše hotovým kabelem připojit k programátoru. Pro nemožnost přehození kabelu jsem použil header, u kterého toto nehrozí. Jelikož na programátoru je jen 4pinová lišta, tak jsem tento kabel na programátor přilepil, abych tak zabránil poškození jak programátoru, tak desky s plošným spojem.



Obrázek 4.13: Elektronické schéma interfazu

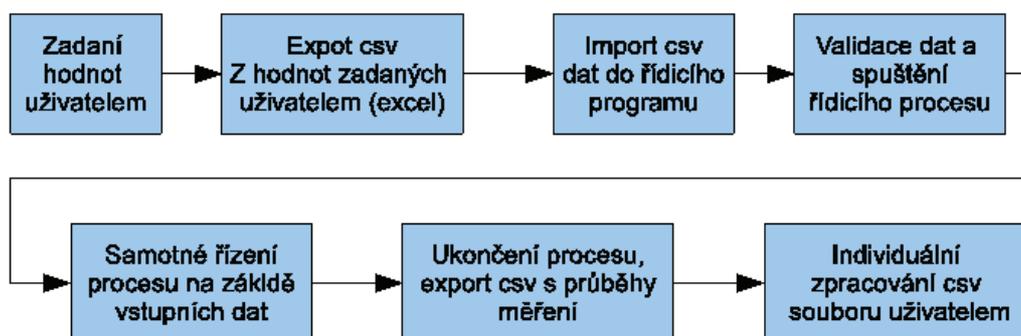


Obrázek 4.14: Navržená deska pološného spoje

Kapitola 5

Příprava a zpracování dat pro měření

Celý proces je silně závislý na laborantovi. Na jeho prvotním importu dat, konečném exportu a následném zpracování dat. Celý proces je popsán na následujícím grafu:



Obrázek 5.1: Schéma cesty dat v procesu

5.1 Zadání hodnot uživatelem

K zadání těchto parametrů jsem využil možností programu Excel. Tento program dokáže generovat csv soubor a dá se v něm udělat formulář, který bude kontrolovat každou hodnotu, kterou zadá uživatel.

5.1.1 Specifikace vytváření vstupních dat

Vstupními daty pro systém je soubor hodnot ve formátu csv. Tento formát je univerzálním formátem a díky tomu dokáže spolupracovat s velkou řadou programů. Jedním z nich je i LabView, který ho dokáže jak číst, tak i vytvořit. Požadavky pro vygenerování tohoto souboru byly zajistit jednoduchými technikami, aby soubor byl vygenerovaný bez chyb a mohl jej program v LabView bez problémů zpracovat. Je tedy nutné zabránit uživateli zadávat do polí nesmyslné hodnoty a udělat určitou autentizaci souboru. Touto autentizací ověřit, zda byl soubor vytvořen pro účel měření a zabránit tak importu souborů, které neobsahují hodnoty nutné pro řízení procesu.

5.1.2 Vytvoření formuláře a rozpoznání chyb zadaných uživatelem

K vytvoření uživatelského prostředí jsem použil program Excel a jeho možnost programování v VBA, nebo-li maker.

Požadovaný časový průběh hodnot				Nastavení proudu vybíječem				Poruchy			
Čas[s]	Akce	Regulované napětí	Regulovaný proud					Nabíjení		Vybíjení	
								MAX	MIN	MAX	MIN
0	Nabíjení	6	0	X	X	X	X	1.8 Ω			
3	Nabíjení	8	0	X	X	X	X	U _{max} 90 V	1.8R / 90V -> MAX: 0,6R ; 90V ; 150A	Porucha	
6	Vybíjení	0	3	X	X	X	X			Proud[A]	10 2 2 1
7	Vybíjení	0	3	X	X	X	X	1.2 Ω	U _{max} 60 V	Napětí[V]	13 7 13 3
8	Nabíjení	11	0	X	X	X	X	U _{max} 60 V	1,2R / 60V -> MAX: 0,4R ; 60V ; 150A	Teplota 1[°C]	10 2 2 -3
				X	X	X	X			Teplota 2[°C]	13 -5 -5 -5
				X	X	X	X	0.9 Ω	U _{max} 45 V	Pozn: hodnoty poruch vymezují marginální hodnoty při nabíjení a vybíjení baterii, jednotlivé poruchy se dají po importu do programu vypnout či zapnout	
				X	X	X	X	U _{max} 45 V	0,9R / 45V -> MAX: 0,3R ; 45V ; 150A	Nastavené maximální hodnoty	
				X	X	X	X	0.6 Ω	U _{max} 30 V	Maximální proud [A]	150
				X	X	X	X	U _{max} 30 V	0,6R / 30V -> MAX: 0,2R ; 30V ; 150A	Maximální napětí [V]	15
				X	X	X	X	0.3 Ω	U _{max} 15 V	Vypínací proud při nabíjení [V]	7
				X	X	X	X	U _{max} 15 V	0,3R / 15V -> MAX: 0,1R ; 15V ; 150A	Vygeneruj	
				X	X	X	X	0.2 Ω	U _{max} 10 V		
				X	X	X	X	U _{max} 10 V	0,2R / 10V -> MAX: 1/15R ; 10V ; 150A		

Obrázek 5.2: Obrazovka uživatelského rozhraní

Jedná se o jednoduchou tabulku v Excelu, kdy je každá buňka, kterou může uživatel vyplnit chráněná tak, aby mohl vyplnit jen povolené hodnoty. První čtyři sloupce jsou stěžejní pro řízení celého procesu. Jedná se totiž o časový průběh všech hodnot, které uživatel požaduje. Vysvětlení jednotlivých hodnot:

Jméno hodnoty	Význam
Čas	je čas od začátku procesu. Právě v tento čas se nastaví na řídicí prvky hodnoty, které jsou na řádku s tímto časem
Akce	určuje zda se v intervalu [čas(tento řádek), čas(tento řádek+1)] bude stanice nabíjet či vybíjet
Regulované napětí	určuje napětí v intervalu [čas(tento řádek), čas(tento řádek+1)] napětí, které má být na baterce, platí jen pro nabíjení, pokud je akce vybíjení, musí být tato hodnota 0 a v řídicím programu na ni není brán ohled.
Regulovaný proud	určuje proud v intervalu [čas(tento řádek), čas(tento řádek+1)], který má být na baterce, platí jen pro vybíjení, pokud je akce nabíjení, musí být tato hodnota 0 a v řídicím programu na ni není brán ohled.

Tabulka 5.1: Význam hodnot v uživatelském rozhraní

Dalším ovládacím prvkem je skupina option buttonů, které určují jak uživatel před vybíjením nastavil vybíječ. Výběrem jednoho z option buttonů se nám omezuje proud a napětí, kterým můžeme vybíjet. Při zvolení určitého option buttonu se nám ihned hodnota přenesse do položky Maximální napětí a Maximální proud a tyto hodnoty nám pak omezují hodnoty ve sloupcích regulované napětí a regulovaný proud a také hodnoty pro maximální napětí a proud v poruchách.

Posledním místem, do kterého uživatel může zasahovat jsou hodnoty poruch.

5.1.3 Ochrana před uživatelem

Veškeré hodnoty vyplňované uživatelem mají dvojitý jistění. Prvním jistěním je možnost přímo v Excelu, kdy jedné buňce povolíme jen určitý rozsah čísla. Tento rozsah může být definován buď některou z ostatních buněk nebo přímo číslem. Tento druh filtru se nazývá "ověření dat". Ovšem toto jedno zabezpečení není zcela dokonalé. Jde o to, že pokud je jedna buňka omezena jinou buňkou, která je také editovatelná uživatelem, může se pak stát, že uživatel zadá nejdřív buňku omezenou a poté změní hodnotu meze. Ovšem hodnota omezené buňky se nepřepočítá. Proto je zde druhé jistění hodnot a to přímo při

generování csv souboru.

Ke generování csv souboru jsem použil makra. V makrech jsem ověřil nejdříve veškeré hodnoty proměnných a poté jsem nakopíroval hodnoty do druhého listu. První řádek hodnot na druhém listu jsou 3 náhodná čísla různá od nuly. A pak následuje jejich součin. Je to další zabezpečení správného importu. Tuto jednoduchou operaci provádí i program, do kterého je soubor importován a dochází tak k jednoduchému ověření dokumentu, že se jedná o dokument s daty pro měření.

Po ověření všech hodnot a nahrání těchto hodnot do druhého listu, přijde oznámení uživateli, kolik řádků souboru bylo zpracováno. Jde o to, že vždy musí být vyplněn celý řádek prvních 4 sloupců. Pokud celý řádek vyplněn není, tak na prvním řádku, který není celý vyplněn, skončí import. Proto program zahlásí, kolik řádků načetl. Po potvrzení této hodnoty je uživatel vyzván k tomu, aby zadal cestu, kam chce soubor uložit. Následně je spuštěn příkaz pro vygenerování csv souboru z jednoho listu.

Kapitola 6

Program pro řízení celé soustavy

6.0.4 Něco málo o LabView

Program pro řízení celé soustavy byl naprogramován v programovacím prostředí LabView. Toto prostředí není klasické prostředí, kde by se psal kód programu, ale jedná se o speciální způsob programování, který vyvinula firma National Instruments. Tento programovací "jazyk" nebo asi lépe tato programovací technika je založena na data flow, tedy na toku dat [1]. Jedná se o to, že každý prvek v tomto programovacím prostředí je reprezentován "kostičkou", která má vstupy a výstupy. Každá tato kostička je vždy vykonána teprve až dostane potřebná data na její vstupy. Celé programování je založeno na tom, že jsou zde funkce reprezentované kostičkou, kterou umísťujeme na prostor, kde nám vzniká program. Tyto jednotlivé kostičky spojujeme virtuálními vodiči a tím určujeme souvislosti jednotlivých kostiček (funkcí). Jednotlivé kostičky jsou umístěny v panelu diagram, ale pak zde máme ještě panel s uživatelským rozhraním. Panel je čelní obrazovka, neboli okno, které uživatel vidí a může přes něj celý program ovládat. Na tento panel umísťujeme vizuální prvky, které se nám poté v diagramu promítnou jako kostičky a nadále s nimi pracujeme jako s kostičkou v diagramu, kde tento prvek parametrizujeme (zapisujeme do něj a čteme z něj)

6.0.5 Představa o programu

Celý program má 3 módy ovládání. Prvním módem programu je manuální ovládání. V tomto ovládání uživatel ručně (klikem) zasahuje do celého programu a hned ovládá celý systém. Má možnost nastavit stav stykačů, nastavit velikost nabíjecího napětí a vybíjecího proudu. Jedná se tedy o primitivní řízení, jako by uživatel před sebou měl

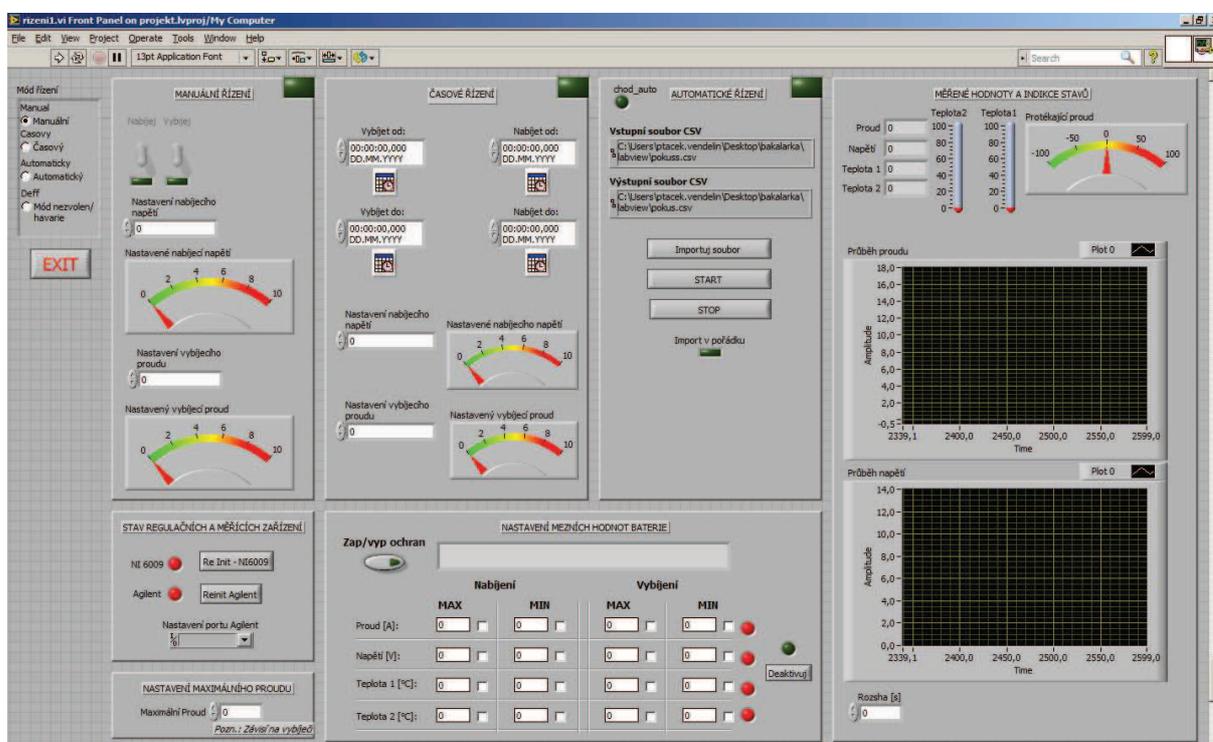
dva potenciometry a dva vypínače. Druhým módem je časový mód. Jedná se o mód kdy uživatel nastaví, v jaké době chce nabíjet a v jaké době chce vybíjet a následně program vykoná akce tak, jak uživatel zadal. S tím, že zadá i vybíjecí proud a nabíjecí napětí, ale tyto dvě hodnoty zůstávají po celý proces nabíjení/vybíjení stejné. Posledním módem programu je plně automatický režim. Tento režim vyžaduje vstupní csv soubor, na jehož základě je celý proces ovládán. Tento soubor obsahuje čas a hodnoty veličin. Program bere jednotlivé řádky CSV souboru a dle nich nastavuje výstupy systému tak, aby korespondovaly se souborem vygenerovaným uživatelem. Ke generování souboru je vytvořená utilita v excelu, o které jsem psal výše. Pokud zvolíme jeden z režimů, ostatní se odpojí a jejich tlačítka zšednou a nelze je ovládat, lze ovládat jen zvolený mód. Pokud dojde k výpadku z některých periférií (měřič napětí nebo vzdálené vstupy a výstupy NI 6009), dojde k přepnutí módu do default módu. v tomto módu nemůžeme nic ovládat a systém je v chybě, po opravení chyby reinitování špatné periferie, můžeme znovu nastavit mód a dále pokračovat v práci.

Program také obsahuje nutné nastavení poruch, kdy uživatel zvolí pro danou baterii poruchové stavy a následně program tyto poruchové stavy hlídá a při vyskytnutí poruchy je celý program pozastaven. Dále program obsahuje indikační prvky veškerých měřených a řízených veličin.

6.0.6 Přední panel

Přední panel by se dal rozdělit do 6-ti sekcí. První z nich jsou ovládací prvky manuálního ovládání, následně pro časové a automatické ovládání. Dále je v pravém horním rohu rozskok těchto tří způsobů ovládání. Po zvolení jednoho způsobu ovládání se u panelu právě tohoto ovládání rozsvítí zelené signalizační světlo, které naznačuje, který ze způsobů ovládání je zvolen. Pokud je zvolen jeden způsob ovládání, pak ovládací prvky ostatních způsobů ovládání jsou vyšedlé a nedá se na ně klikat. Další sekce je sekce s měřenými hodnotami. Tyto hodnoty se zobrazují v pravé části panelu. Zobrazují se zde aktuální hodnoty měřených veličin. Dále zde máme dva grafy, které nám ukazují průběh napětí a proudu. Tyto průběhy mají nastavitelný rozsah osy X a to textboxem pod nimi. Další důležitou sekcí je sekce poruch. Je to sekce kde uživatel zadává mezní hodnoty teplot proudů a napětí pro jednotlivé činnosti a to nabíjení nebo vybíjení. Dále je u každé hodnoty checkbox, kterým se tato porucha povolí nebo zakáže. Dále jsou zde signalizační prvky jednotlivých poruch. Po nastání poruchy se celý proces nabíjení nebo vybíjení zastaví. A čeká se na zásah uživatele, který tuto poruchu odblokuje stisknutím na tlačítko

deaktivuj. Dále se dají veškeré poruchy vypnout a to spínačem Zap/vyp ochran. Při rozepnutí tohoto spínače dojde k deaktivaci ochran a vypíše se hlášení červeně "POZOR program běží bez ochran". Toto hlášení se vypíše, jelikož chod bez poruch může mít destruktivní následky a každá ochrana proti těmto následkům by měla být zapojena. Poslední sekci je sekce s připojenými periferiemi. Jedná se o periferie, které jsou důležité pro chod programu a to vzdálené vstupy a výstupy NI 6009 a multimetr Agilent. Stav těchto připojených zařízení je naznačen virtuální diodou a je zde možnost reinitializace. Když spadne komunikace, ať už s NI6009 nebo s multimetrem, program nespadne, ale nahlásí jen chybu a uživatel může zkusit znovu navázat komunikaci s daným zařízením.



Obrázek 6.1: Přední panel

6.0.7 Manuální a časový režim

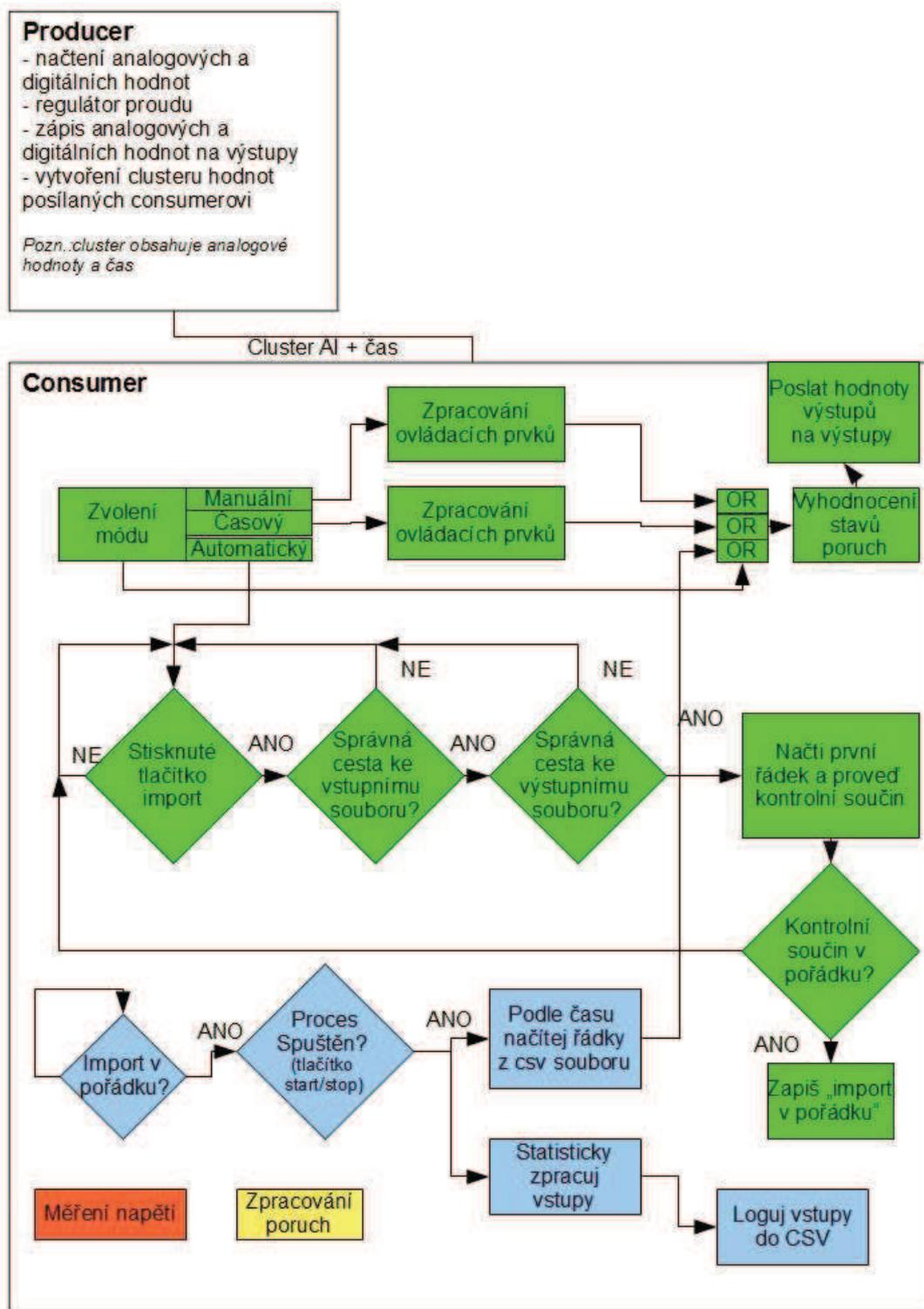
Tyto dva režimy jsou velice jednoduché. Jejich výstup záleží na aktuálním nastavení uživatele (u manuálního režimu) a na čase, který uživatel nastaví (u časového režimu).

6.0.8 Automatický režim

Pro spuštění automatického režimu je nutné zadat vstupní soubor csv. Tento soubor musí být generovaný výše popsaným programem v excelu. Poté co uživatel zadá tento soubor a také zadá adresu výstupního souboru, může dát tlačítko import. Pokud uživatel nezadá správný soubor, program vypíše chybu. Po načtení tohoto souboru se provádí validace souboru, zdali má soubor na prvním řádku "kontrolní násobek" a následně se načítají hodnoty poruch a načítají se do sekce programu poruchy. Poté program čeká na to, až uživatel spustí tento režim. Po spuštění program zaznamená aktuální čas, a pak od tohoto času počítá čas započatého procesu. Podle času tohoto procesu potom program čte hodnoty csv souboru a zapisuje je na proměnné, které řídí hodnotu proudu, hodnotu napětí a také stavy stykačů.

6.0.9 Program a jeho struktura

Celá struktura programu se skládá ze dvou částí. Celý program je totiž navržen jako producer consumer. Producer je první while smyčka. V této smyčce je zajištěno čtení a zápis do/z NI 6009. Pro sběr dat ze vstupů jsou vstupy nahrány do pole a poslány do druhé smyčky consumer ke zpracování. Zde proběhne zpracování statistické, kde je pole seřazeno a 10% hodnot pole shora a zespod je smazáno a následně je vrácen průměr zbylých hodnot. Tyto hodnoty se pak dále zpracují.

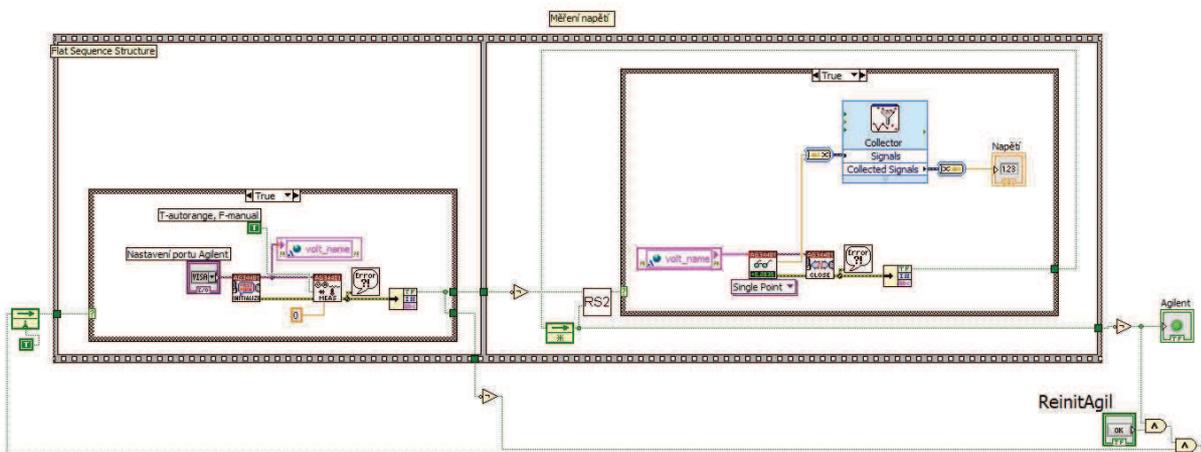


Obrázek 6.2: Blokové schéma programu

Barevně jsou v diagramu rozlišeny paralelně pracující virtuální přístroje.

6.0.10 Měření napětí

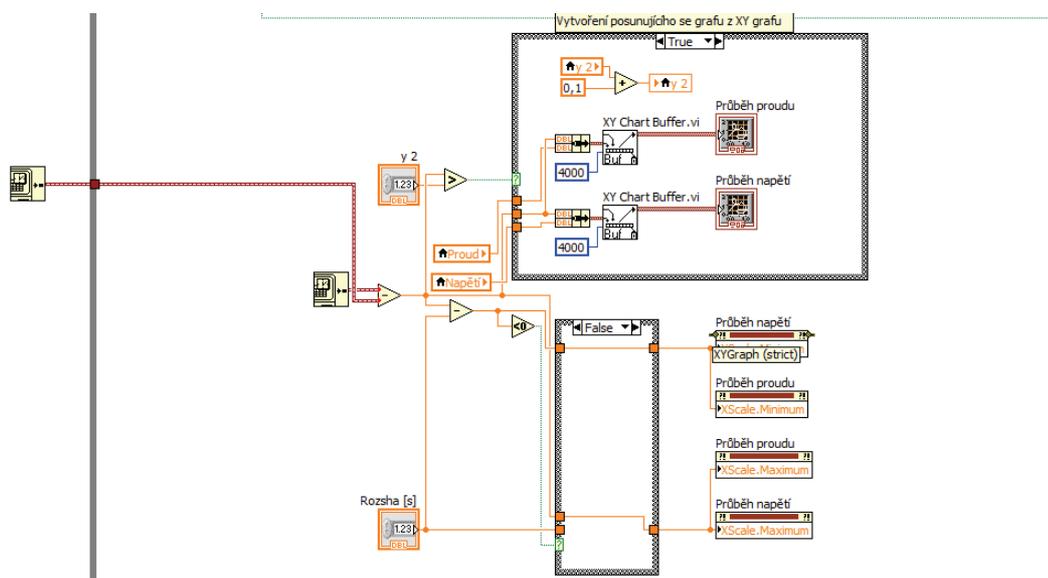
K měření napětí je využit Agilent 34410A, který má USB interface k připojení k počítači. Po spuštění prostředí LabView jsem zadal najít zařízení a následně stáhnout potřebné ovladače. Poté přímo v nabídce virtuálních nástrojů jsem našel potřebné virtuální nástroje k parametrizování tohoto zařízení. Na obrázku jde vidět diagram v LabView, který ovládá tento měřicí přístroj. Tento diagram je naprogramován tak, aby nejdříve proběhla inicializace a poté se začalo měřit. A inicializace už dalším průchodem programu neprobíhala. Tato inicializace proběhne jenom na startu program a nebo pokud komunikace spadne, tak uživatel může tuto inicializaci pustit sám a to tlačítkem "Agilent re-init".



Obrázek 6.3: LabView diagram pro měření napětí

6.0.11 Modifikovaný XY graf

Žádný z nabízených zobrazovačů hodnoty mi nevyhovoval tak, jak jsem chtěl. Chtěl jsem mít něco jako pomalý osciloskop, kdy by se nastavil časový rozsah tohoto indikátoru a on stejně jako na osciloskopu zobrazoval hodnotu jakou jsem chtěl. Takový indikátor ovšem není, tak jsem jej musel vytvořit ze standardních prvků.

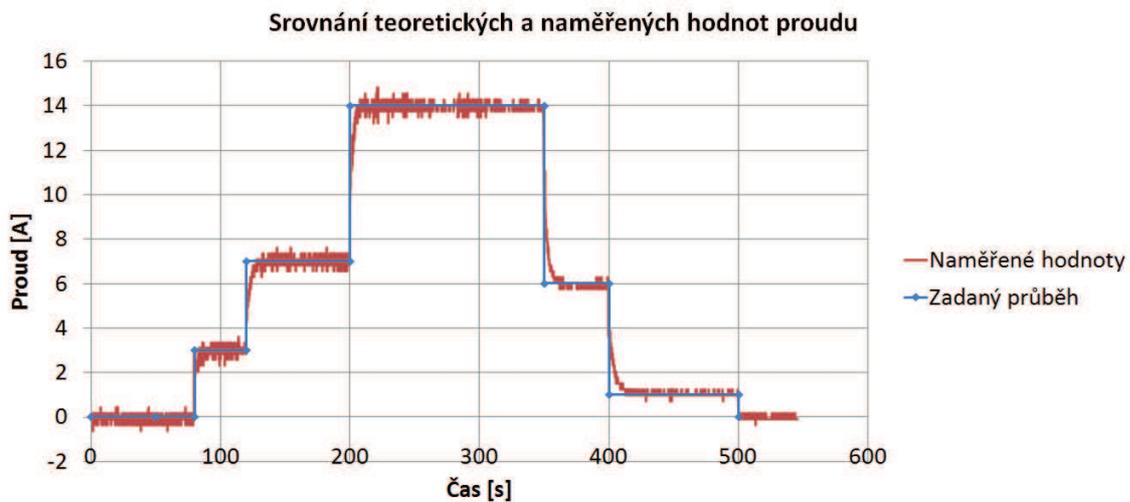


Obrázek 6.4: LabView diagram pro vlastní indikátor

Celý princip tohoto indikátoru je založen na dvou věcech. První z nich je, že jeho absolutní hodnota rozsahu osy X je stále stejná. A podle času se maximum a minimum osy pohybuje takto: $X_{min} = \text{aktuální čas} - \text{rozsah}$, $X_{max} = \text{aktuální čas}$. Tím zajistím, že se mi osa posouvá. Druhou věcí je, že XY graf je statický a vykresluje se na základě clustru dvou polí. K tomu abychom mohli postupně načítat další a další hodnoty, musel se před objekt grafu vložit buffer. Jelikož program je velice rychlý a každým průchodem programu se zapíše další hodnota, docházelo často k přetékání bufferu. To jsem vyřešil tak, že se diagram pro načítání hodnot do grafu spouští pouze každou desetinu sekundy, což zajišťuje dostatečně plynulý graf a buffer zajišťuje dostatečný prostor pro předcházející hodnoty.

6.0.12 Měření

Bylo provedeno jedno měření, kterým jsem finálně ověřil funkčnost celého programu. Výstupem byl csv soubor, který jsem následně zpracoval v Excelu. Regulace napětí ještě není k dispozici, jelikož nabíječ se dá jen sepnout nebo vypnout stykačem. Ale pro regulaci napětí je vše připraveno. Jelikož tedy napětí nelze regulovat, zpracoval jsem pouze graf a srovnání teoretického a naměřeného proudu.



Obrázek 6.5: graf srovnání předepsaného proudu s naměřeným

Z grafu lze poznat, že měřený proud takřka kopíruje proud nastavený. Měřený proud LEM sondou je rozkmitaný a k systému bude navrženo jiné měření proudu. Buď se nechá stávající měření a přidá se filtr, nebo se měření proudu zcela změní.

Kapitola 7

Závěr

V rámci bakalářské práce bylo navrženo řízení pro stávající nabíjecí/vybíjecí stanici. První částí práce bylo navržení řízení vybíječe, s tímto vybíječem jsem se seznámil, navrhl řízení pomocí PWM signálu a následně realizoval a to pomocí mikrokontroléru STM8s. Dalším úkolem bylo vytvořit řídicí program pro všechny prvky této soustavy. Tento řídicí program jsem vytvořil ve vývojovém prostředí LabView. Nakonec jsem tento program otestoval na experimentálním měření a tím ověřil jeho funkčnost.

Literatura

- [1] J. Havlíček, J. Vlach, M. Vlach, V. Vlachová: *Začínáme s LabVIEW .BEN* – technická literatura, Praha, 2008. ISBN 978-80-7300-245-9.
- [2] M. Kmínek: *Testovací zařízení pro měření baterií*. ČVUT fel, Praha, 2010
- [3] STMicroelectronic®: *Aplikační poznámka AN3332: Generating PWM signals using STM8S-DISCOVERY*. (Květen 2012).
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/APPLICATION_NOTE/CD00296680.pdf
- [4] STMicroelectronic®: *Aplikační poznámka AN3332: AN3280: Displaying variable voltage on a bar of LEDs using STM8S-DISCOVERY*. (Květen 2012).
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/APPLICATION_NOTE/CD00285727.pdf
- [5] STMicroelectronic®: *Uživatelská příručka: UM0817: STM8S-DISCOVERY*. (Květen 2012).
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/USER_MANUAL/CD00250600.pdf
- [6] STMicroelectronic®: *Uživatelská příručka: UM0834: Developing and debugging your STM8S-DISCOVERY application code*. (Květen 2012).
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/USER_MANUAL/CD00250600.pdf
- [7] STMicroelectronic®: *Datasheet: DS5855: Access line, 16 MHz STM8S 8-bit MCU, up to 32 Kbytes Flash, integrated EEPROM, 10-bit ADC, timers, UART, SPI, I²C*. (Květen 2012).
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00200092.pdf

- [8] STMicroelectronic®: *Referenční manuál:RM0016: STM8S and STM8A microcontroller families*. (Květen 2012).
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/REFERENCE_MANUAL/CD00190271.pdf
- [9] LEM®: *Datasheet: Current Transducer HASS 50..600-S*. (Květen 2012).
http://www.lem.com/docs/products/hass_e_rev14.pdf
- [10] National Instruments®: *Produktová stránka NI USB-6009*. (Květen 2012).
<http://sine.ni.com/nips/cds/view/p/lang/en/nid/201987>
- [11] Agilent Technologies: *Produktová stránka 34410A Digital Multimeter, 6[?] Digit High Performance*. (Květen 2012).
<http://www.home.agilent.com/agilent/product.jsp?nid=-33228.536908384.00&cc=CZ&lc=eng>
- [12] STMicroelectronic®: *Datasheet: DS0437: Precision temperature sensors*. (Květen 2012).
<http://www.st.com/internet/analog/product/63680.jsp?referrer=70071840>
- [13] Eprona a.s.: *Produktová stránka:Spínané nabíječe řady HFH-A*. (Květen 2012).
<http://www.eprona.cz/cz/hfha.html>
- [14] F. Peter a spol.: *OpenModelica User Guide*. PELAB, Linköping University, Sweden, 2006
<http://build.openmodelica.org/omc/builds/windows/releases/1.4.2/doc/OpenModelicaUsersGuide.pdf>

Příloha A

Obsah přiloženého DVD

K této práci je přiloženo DVD, na kterém je veškerá elektronická dokumentace se softwarem a zdrojovými kódy k této práci

- datasheet: datasheety, které byly použity v této práci
- generator_csv: excelový soubor pro generování csv souborů
- OpenModelica: software a zdrojové kódy, které sem použil k modelování
- osciloskop: data z měření na osciloskopu
- plosny_spoj: plošný spoj s jeho schématem a softwarem pro otevření a editaci
- program_LabView: řídicí program naprogramovaný v LabView
- soubory_excel: zpracované hodnoty měření v Excelu
- stm8_program: Zdrojové kódy a vývojové prostředí k STM8
- pdf verze této práce

Pozn.: v každé složce je ReadMe.txt s popisem co se ve složce nachází.