Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Tester vybíjecích charakteristik primárních článků

Plzeň 2020 Martina Janečková

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Martina JANEČKOVÁ**
Osobní číslo:	**A17B0460P**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Výpočetní technika**
Téma práce:	**Tester vybíjecích charakteristik primárních článků**
Zadávající katedra:	**Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s vlastnostmi a s vybíjecími charakteristikami primárních článků.
2. Navrhněte celkovou koncepci testeru, který bude mít následující parametry:
   a) Možnost volby nastavení profilu vybíjení článku.
   b) Možnost záznamu a následného vyhodnocení vybíjecí charakteristiky.
   c) Ovládání jednoduchým grafickým rozhraním, včetně grafického zobrazení vybíjecí charakteristiky.
   d) Maximální vybíjecí proud v řádu jednotek ampér.
   e) Maximální napětí článku 5 V.
3. Zvolte vhodnou platformu a navrhněte další potřebné technické vybavení testeru.
4. Vytvořte příslušné programové vybavení testeru.
5. Celé zařízení podle možnosti realizujte.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Dr. Ing. Karel Dudáček**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **7. října 2019**
Termín odevzdání bakalářské práce: **7. května 2020**

Západočeská univerzita v Plzni
Fakulta aplikovaných věd

L.S.

Radová

Doc. Dr. Ing. Vlasta Radová
děkanka

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 15. října 2019

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 20. července 2020

Martina Janečková

Abstract

The text focuses on problematics of primary cells capacity and their discharge characteristics. In the first part is introduction to the topic of galvanic cells and theirs individual types. Main topic of the text is desription of hardware and software design for the primary cell discharging device. Software includes PC user interface and devices firmware. Hardware includes power and regulation circuits.

Abstrakt

Tato práce se zabývá problematikou kapacity primárních článků a jejich vybíjecími charakteristikami. V první části práce se nachází úvod do problematiky galvanických článků a specifik jednotlivých typů. Hlavní obsahem je popis vývoje hardwarového a softwarového příslušenství vybíječe primárních článků. Software zahrnuje uživatelské rozhraní pro PC a firmware vybíječe. Hardware zahrnuje výkonovou a regulační část vybíječe.

Poděkování

Chtěla bych poděkovat svému vedoucímu bakalářské práce Dr. Ing. Karlu Dudáčkovi za odborné vedení, za pomoc a rady při zpracování této práce.

Obsah

Seznam zkratek a symbolů

PC - Personal computer
Q_{bat} - Elektický náboj baterie
I_{bat} - Vybíjecí proud baterie
I_{max} - Maximální vybíjecí proud baterie daný napětím článku
PWM - Pulse Width Modulation (Pulzní Šířková modulace)
AD - Analog to Digital (přechod z analogového na digitální signál)
MOSFET - Metal Oxide Semiconductor Field Effect Transistor
MCU - MicroController Unit (jednočipový počítač)
f_{out} - Výstupní frekvence PWM signálu
f_{clk} - Frekvence vnitřních hodin MCU
ICRn - Registr určující periodu nosného signálu PWM
OCRn - Registr určující šířku pulzu PWM
ADC - Hodnota změřená AD převodníkem
JDK - Java Development Kit
USB - Universal Serial Bus (universální sériová sběrnice)

1 Úvod

V dnešní době jsou mobilní elektrická zařízení součástí každodenního života a společně s nimi se nutně stali součástí každodenního života baterie potřebné k jejich napájení. Jejich nákup je ale mnohdy složitý a například při nákupu jednorázových baterií nemá zákazník kromě ceny a data expirace k dispozici žádné vodítko, pomocí kterého by mohl zhodnotit kvalitu kupovaného článku. Zákazník tak čelí velmi nejistému nákupu.

Ani průzkum internetu zákazníkovi většinou nepomůže. Je zde velmi malý počet testů kapacit baterií a většina z testovaných baterií se už často ani nevyrábí, nebo nejsou v České republice k sehnání. Stránky výrobců baterií také moc informací nenabízejí a i kdyby ano, je otázkou, jak moc by se těmto informacím dalo věřit a také, jaká je konzistentnost výroby, zda se mezi sebou neliší i články od jednoho výrobce.

Cílem této práce je navrhnout a vytvořit tester vybíjecích charakteristik primárních článků, který se bude dát jednoduše ovládat z PC. Díky tomuto zařízení půjde články jednoduše vybít za volitelného vybíjecího proudu a získat tak jejich kapacitu a průběh vybíjecí charakteristiky.

2 Galvanické články

Historii galvanických článků začal psát italský fyzik Alessandro Volta v roce 1800. Sestavil článek skládající se z kladné měděné a záporné zinkové elektrody ponořené do roztoku kyseliny sírové. Vznikl tak první stálý zdroj elektrického proudu. Funkce galvanického článku je na principu přeměny chemické energie na energii elektrickou. Nejprodávanějšími a nejlépe dostupnými, i díky nižší ceně, jsou primární články. Slouží pouze na jedno použití a po vybití se nedají znovu nabít. Bohužel značně znečišťují životní prostředí. Jejich velmi dobrou náhradou jsou články sekundární, které lze po použití opakovaně nabíjet.[1]

2.1 Primární články

Jako primární články se označuje skupina chemických zdrojů elektrické energie, u kterých při vybíjení dochází k chemické reakci a přeměňují chemickou energii na energii elektrickou. Mají omezené množství reaktantů a vybíjením článků se reaktanty spotřebovávají na produkty, které už nelze převést zpět na původní reaktanty. Jsou určeny pro jednorázové použití a nedají se opakovaně nabíjet. Nejznámějšími primárními články jsou tužkové baterie. [2]

Obsahují jedovaté látky, tudíž jsou velmi neekologické a na výrobu je spotřebováno o dost víc energie, než článek vydá v podobě elektrické energie. Mezi výhodami primárních článků je odolnost vůči samovybíjení. Obvykle jsou uzavřené, mechanicky pevné a odolné vůči otřesům. Mezi funkční části patří kladná a záporná elektroda, elektrolyt a obal.

Primární články se liší dle napětí, tvaru a použité chemie. Podle typu anody se primární články dělí na tři hlavní skupiny: zinkové, lithiové, hořčíkové. Mezi nejrozšířenější skupinu patří články se zinkovou anodou, které lze ještě dále dělit. Nejrozšířenější články se zinkovou anodou jsou suché a alkalické. Dalším velmi rozšířenou technologií jsou lithiové články.

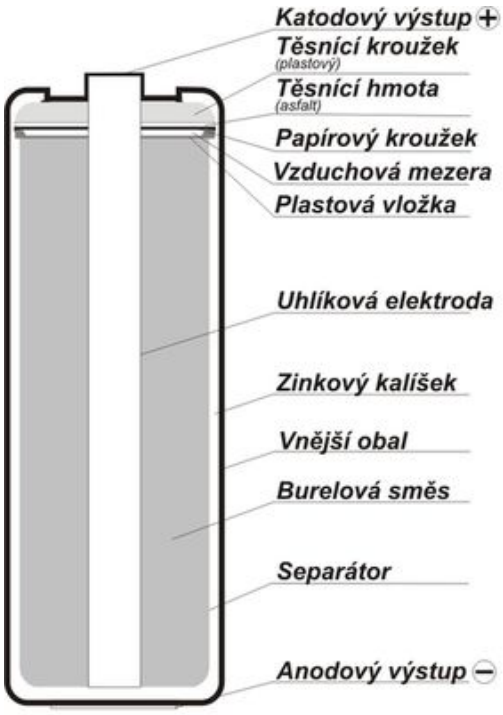
2.1.1 Články Leclanchova typu

První suchý článek sestavil fyzik Georges Leclanché roku 1866. Název suchý článek je odvozen od elektrolytu, který měl gelovitou strukturu. Jako anoda byl použit zinek ve formě kalíšku. Elektrolyt uvnitř kalíšku byl původně vodný roztok chloridu amonného (salmiak), který byl zahuštěn škrobem.

Katodu tvořila uhlíková elektroda. Uhlíková katoda byla obalena tzv. depolizátorem, který v tomto případě byl oxid uhličitý, též nazývaný burel. U této technologie docházelo při vybíjení ke vzniku volné vody a tím bylo zapříčiněno vytékání elektrolytu.

Články Leclanchova typu jsou označovány jako zinkouhlíkové, nebo také jejich vylepšená verze zinkochloridové (podle vnějšího vzhledu elektrod) a koncepce se od jejich předchůdce zásadně neliší. Anodu opět tvoří zinkový kalíšek, který zároveň tvoří i vnější obal. Elektrolyt taktéž tvoří chlorid amonný, ale je nasáknut do kladné elektrody, kterou tvoří prášková směs uhlíku a oxidu manganičitého. Mezi obalem a kladnou elektrodou se nachází separátor ze savého papíru, který slouží pro zabránění přímého kontaktu mezi elektrodami. V průběhu vybíjení se zinek spotřebovává a vniká voda. Pokud obal není dostatečně silný, může se zinek proděravět a elektrolyt vyteče.

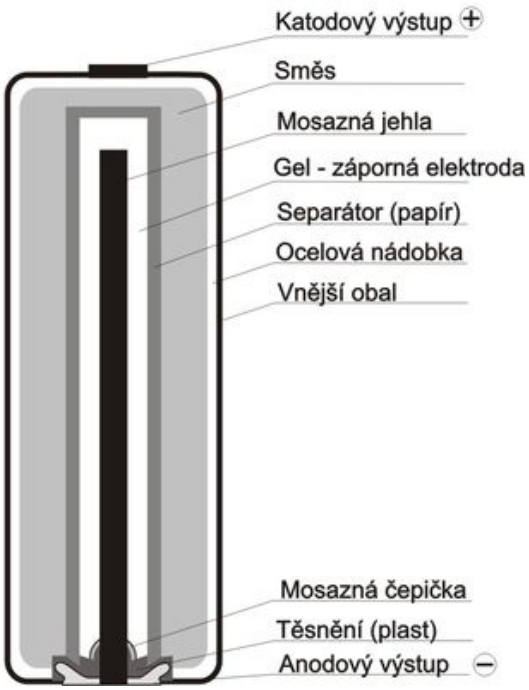
Zinkochloridové baterie mohou být také vyrobeny z lepších chemikálií a funkci elektrolytu u nich pak zastává roztok chloridu zinečnatého. Při vybíjení se v článku netvoří voda, ale naopak je voda při vybíjení spotřebovávána a článek je tak ke konci své životnosti suchý. Důležité je, aby obal poskytoval dostatečné těsnění a nemohlo tak dojít k předčasnému vyschnutí článku.



Obrázek 2.1: Průřez zinkochloridovým článkem [3]

2.1.2 Alkalické baterie

Používaná konstrukce alkalických primárních článků vznikla již v minulém století. Snahou bylo potlačit negativní vlastnosti zinkochloridových článků. Alkalické baterie přinášejí dobrý výkon a přijatelnou cenu především pro zařízení s nízkým odběrem (budíky, dálkové ovladače, hračky a podobně), kde se nemusejí často měnit. Materiály obou elektrod, které se podílejí na proudotvorné reakci zůstaly stejné, ale zásadně se změnilo vnitřní uspořádání článku 2.2. Fungují na chemické reakci mezi oxidem manganičitým na katodě a zinkem na anodě, který je v podobě slisovaného prášku v elektrolytu umístěn ve středové části válcového pouzdra. Katoda a anoda jsou od sebe odděleny membránou. Elektrolytem v alkalických bateriích je hydroxid draselný, který je proti vnějšímu pokovenému ocelovému obalu málo agresivní, tudíž nedochází k proděravění článku kvůli reakci s hydroxidem. Další výhodou tohoto uspořádaní článku je větší plocha elektrod, a tím i větší získaná energie v porovnání se stejně velkým zinkochloridovým článkem. Mají jmenovité napětí 1,5V a zvládnou zatížení až kolem 700mA bez významného zahřívání. [4]



Obrázek 2.2: Průřez alkalickým článkem [5]

2.1.3 Lithiové primární články

Lithiové primární články patří na trhu mezi velké konkurenty alkalických článků. Vydrží velmi nízké teploty, mají vyšší kapacitu a schopnost dodávat vyšší proud ve srovnání s alkalickými články. Jejich životnost bývá až 20 let, protože mají velmi nízké samovybíjení. Používají se tedy především v zařízeních, kde je potřeba jejich dlouhá životnost. Jejich jmenovité napětí se pohybuje od 1,5V do 3,7V. Anodu tvoří kovové lithium (alkalický kov) nebo jeho sloučenina. Materiál katody závisí na použití, nejčastěji se používá oxid manganičitý. Vzhledem k tomu, že lithium je velmi reaktivní, musí být při výrobě všechny operace prováděny v boxech s ochrannou atmosférou. Prvky použité k výrobě nesmí obsahovat stopy vlhkosti nebo jiných látek reagujících s lithiem. Články musí být dokonale hermetické, aby do nich nemohla proniknout vlhkost a znehodnotit je.

2.2 Sekundární články

Galvanické články, které lze opakovaně vybít a následně opět nabít se nazývají akumulátory nebo sekundární články. Jedním z jejich důležitých parametrů je počet pracovních cyklů, tzn. kolikrát lze článek vybít a znovu nabít. V dnešní době jsou to stovky až tisíce cyklů. Během nabíjení se článek připojí ke zdroji stejnosměrného napětí. Elektrická energie se v článku přemění na chemickou a následně je článek schopen opět dodávat naakumulovanou energii. [6]

Původní olověný akumulátor navrhl roku 1859 francouzský fyzik Raymond Louis Gaston Planté. Napětí olověných akumulátorů se pohybuje v rozpětí 1,85 - 2,4V. Elektrody jsou z olova a elektrolytem je zředěná kyselina sírová. Při nabíjení se na anodě vytváří oxid olovičitý a na katodě olovo, elektrolyt zhoustne a vytváří se kyselina sírová. Při vybíjení zase naopak elektrolyt řídne, jelikož se spotřebovává kyselina sírová. Ve vybitém stavu je na obou elektrodách síran olovnatý. Stav akumulátoru se zjišťuje měřením hustoty elektrolytu nebo svorkového napětí při zatížení. Nevýhodou olověného akumulátoru je jeho náročná údržba, kdy se musí udržovat neustále v nabitém stavu. Síran olovnatý se, při dlouhodobém stání ve vybitém stavu, stává špatně rozpustný. Při nízkých teplotách kapacita akumulátoru klesá. [7]

3 Parametry článků

Jak bylo napsáno v předchozí kapitole, existuje mnoho různých typů elektrických článků. Pro porovnání vlastností článků lze použít několik ukazatelů, které vycházejí z fyzikálních vlastností.

3.1 Kapacita článků

Kapacita článku udává množství elektrického náboje, které může článek dodat při vybíjení za stanovených podmínek. Kapacita baterie se udává jako integrál z proudu baterie podle času.

$$Q_{bat} = \int I_{bat} \cdot dt \tag{3.1}$$

V praxi se kapacita článků udává v ampérhodinách (Ah), popřípadě se uvádí v miliampérhodinách (mAh). Ampérhodina je jednotka elektrického náboje odpovídající potenciálu dodávat proud 1A po dobu 1 hodiny. Proud je ale závislý na napětí, proto pokud chceme porovnávat kapacity baterií o různém napětí, ideálním řešením je převod na hodnoty ve watthodinách (Wh). Tato hodnota se vypočte vynásobením kapacity v ampérhodinách a jmenovitého napětí.

Množství energie nebo-li skutečná kapacita, které lze z článku získat, je závislé na mnoha faktorech. Konstrukce, aktivní plocha elektrod, čistota materiálů, velikost vybíjecího proudu a provozní teplota během vybíjení mají vliv na množství energie, které je schopen článek poskytnout. Množství energie je dokonce ovlivněno i způsobem vybíjení baterie, který může být přerušovaný nebo stálý. Dalšími faktory jsou doba a podmínky skladování.

3.2 Napětí článků

Napětí baterie je vytvořeno rozdílem elektrických potenciálů (tzn. množstvím energie, které je potřeba k přemístění náboje) mezi kladnou svorkou a zápornou svorkou baterie. Čím větší je rozdíl potenciálů, tím vyšší je i napětí. Napětí článku klesá během vybíjením baterie nebo také vlivem úbytku napětí na vnitřním odporu baterie. Čím více je baterie zatěžována, tím rychleji napětí klesá. Výrobce většinou udává jmenovité napětí, což je průměrné napětí článku během vybíjení.

Baterie se liší napětím v závislosti na velikosti a materiálech, z nichž jsou vyrobeny. Jako životnost baterií se bere časový údaj, při kterém platí provozní podmínky. To u primárních článků znamená pokles napětí na určitou hodnotu, při které už zařízení nebude schopné fungovat.

Například správně navržené zařízení určené k provozu z alkalických baterií by mělo fungovat alespoň do doby, než napětí v bateriích klesne na 0,8 voltu nebo ještě méně. Pokud tomu tak není, zařízení přestane fungovat dříve, než bude baterie zcela vybitá.[8]

3.3 Vnitřní odpor

Vnitřní odpor je důvod vzniku rozdílu při měření napětí elektrického zdroje naprázdno a při měření napětí zatíženého zdroje.

Při navrhování obvodu s baterií často předpokládáme, že baterie je ideálním zdrojem napětí. To znamená, že bez ohledu na to, jak velkou zátěž připojíme k baterii, napětí na svorkách zdroje zůstane vždy stejné.

Realita je ale jiná. Pokud se ke svorkám zdroje připojí zátěž, začne zdroj vykonávat práci a protlačovat tak obvodem elektrický proud, nicméně i zdroj má svůj vnitřní odpor na jehož překonání musí vynaložit práci. Tímto se sníží množství práce, kterou je zdroj schopen dodávat do obvodu a to se projevuje snížením svorkového napětí. [9]

Články jsou konstruovány z materiálů, které mají nenulové odpory. Vnitřní odpor je způsoben strukturálními vadami nebo nepravidelnostmi. Dokonce i kovy vykazují určitý odpor kvůli různým faktorům, jako jsou nečistoty nebo náhodné zahřívání. Kvůli takto vzniklému vnitřnímu odporu se zvyšuje počet srážek iontů s volnými elektrony a pohyb elektronů je tak velmi obtížný. Tudíž zdánlivě nepostřehnutelné odpory všech komponentů v článku jsou v součtu poměrně vysoké a nelze je v některých případech zanedbat. [10]

Většina primárních článků má relativně vysoký vnitřní odpor, což značně omezuje jejich použití na zařízení s nízkým proudem, jako jsou například dálkové ovladače, hračky, kuchyňské hodiny a podobné. V závislosti na hloubce vybití těchto baterií se vnitřní odpor dále zvyšuje. To vysvětluje relativně krátkou dobu provozu při použití běžných alkalických buněk v digitálních fotoaparátech. [11]

4 Součástky k realizaci obvodu

V následující kapitole je shrnuto několik druhů elektrických součástek, které jsou nezbytné pro vytvoření vybíjecího obvodu.

4.1 Operační zesilovač

Operační zesilovač je zesilovač napětí navržený k tomu, aby po jednoduchém doplnění o pasivní součástky zvládl provádět různé úkony. Využití operačních zesilovačů je velmi široké. Nejjednodušším použitím je komparace vstupních signálů. Dalším častým použitím je realizace lineárního zesílení. Ale ve správné konfiguraci zvládne i složitější úkoly, jako je integrace, sčítání nebo použití jako aktivní filtr.

Operační zesilovač zesiluje rozdíl napětí mezi vstupními svorkami. Jeho zesílení je v ideálním případě nekonečné, reálně pak dosahuje stovek tisíc, až jednotek milionů. Zesílení je také limitováno napájecím napětím zesilovače, to totiž určuje maximální výstupní napětí. V případě operačních zesilovačů označovaných, jako *rail-to-rail* je možné dosáhnout na výstupu napětí shodného s napájecím, což je rozdíl proti běžným operačním zesilovačům, které mají maximální výstupní napětí snížené o úbytek uvnitř součástky. Rail-to-rail technologie umožňuje nastavení nulového napětí výstupu bez nutnosti záporného napájecího napětí. [12]

4.2 Tranzistor

Tranzistor je polovodičová součástka určená k řízení nebo spínání elektrického proudu. Tranzistory jsou zodpovědné za současný stav moderní elektroniky a jejich požití je velmi široké, od nanometrových tranzistorů v procesorech po výkonové tranzistory schopné řídit toky výkonů v řádech stovek kilowatt. Podle principu fungování se tranzistory dělí do dvou hlavních skupin a to na bipolární a unipolární. [13]

4.2.1 Bipolární

U bipolárního tranzistoru je proud mezi elektrodami kolektor a emitor řízen proudem mezi elektrodami báze a emitor. Proud tekoucí do kolektoru je výrazně větší než proud tekoucí do báze, proto můžeme mluvit o tom, že bipolární tranzistor zesiluje proud. Zesílení bipolárního tranzistoru, které je často označované jako h_{21}, závisí na konstrukci tranzistoru. Je ale také pravidlem, že čím větší je maximální proud kolektoru, tím nižší je toto zesílení. V dnešní době jsou bipolární tranzistory ve většině aplikací vytlačovány unipolárními tranzistory. [13]

4.2.2 Unipolární

U unipolárního tranzistoru je proud mezi elektrodou drain a source řízen napětím na elektrodě gate. Jelikož do elektrody gate neteče po nabití parazitní kapacity žádný proud, mají unipolární tranzistory nižší nároky na budící obvod a díky tomu vycházejí úsporněji než bipolární. [13]

4.3 Rezistor

Odpory jsou pasivní elektrické součástky, které mají daný specifický odpor. Tento odpor omezuje tok elektronů obvodem. Hodnota odporu může být buď pevně daná nebo proměnná.

U pevných rezistorů se hodnota odporu nedá změnit nebo upravit, lehké změny hodnoty odporu mohou nastat v závislosti na teplotě nebo stáří součástky. Existuje mnoho technologií, kterými se pevné rezistory vyrábějí. Mezi tyto technologie patří například namátkově uhlíkové, metalické nebo drátové. Koncového zákazníka ale většinou zajímají fyzikální parametry rezistorů. Tím nejdůležitějším je elektrický odpor rezistoru, dále v případě některých aplikací je důležitým údajem výkonová zatížitelnost, elektrická pevnost, případně také parazitní indukčnost. Rezistory mají v elektrických obvodech široké uplatnění od slaboproudých aplikací, jako jsou například *pull-down* rezistory, až po výkonové aplikace, kde mohou být využity při regenerativním brzdění motorů. [14]

Proměnných rezistorů je také velká řada druhů. Používají se většinou k plynulému nastavování při ovládání zařízení, jednoduché kalibraci přístrojů, nebo se také používají jako snímače polohy. Dají se dělit podle průběhu změny odporu na lineární nebo logaritmické. Dále se dají dělit podle konstrukce na potenciometry, trimry a reostaty. [15]

4.4 Kondenzátor

Stejně jako v případě rezistoru se jedná o pasivní elektrickou součástku. Kondenzátor zvládne uchovat elektrický náboj. Základní dělení kondenzátorů, které také určuje jejich použitelnost, je podle toho, zda mají jasně danou polaritu, nebo zda na polaritě přivedeného napětí nezáleží.

Polarizované kondenzátory mají ve většině případů výhodu ve větší kapacitě ve srovnání s nepolarizovanými. Jejich nevýhodou je však nebezpečí zničení v případě připojení napětí opačné polarity. Nejčastějšími typy polarizovaných kondenzátorů jsou elektrolytické a tantalové kondenzátory.

Jak již bylo řečeno, nepolarizované kondenzátory mají nižší kapacitu ve srovnání s polarizovanými. Tuto nevýhodu ale vynahrazují možností připojení obou možných polarit napětí, což umožňuje jejich použití ve střídavých obvodech. V některých aplikacích se používají také díky nízkému parazitnímu odporu a indukčnosti. [16]

5 Analýza problému

V prvotním návrhu zařízení je počítáno se třemi hlavními částmi. První částí je výkonový vybíjecí obvod, který se bude řídit pomocí digitálního signálového procesoru. Procesor se bude strat o řízení vybíjecího procesu a komunikovat s aplikací pro vyhodnocení dat a řízení vybíjecího procesu běžící na PC, která bude poslední částí.

5.1 Specifikace požadavků

Na základě zadání bakalářské práce byla vypracována specifikace požadavků. Pro větší přehlednost je specifikace, stejně jako většina dalších kapitol, rozdělena do tří základních částí.

5.1.1 Obvod

Na vybíjecí obvod jsou kladeny dva hlavní nároky. Prvním je požadované napětí článku až 5V. Druhým požadavkem je plynule měnitelný vybíjecí proud až do velikosti jednotek ampér.

5.1.2 Micro Controller

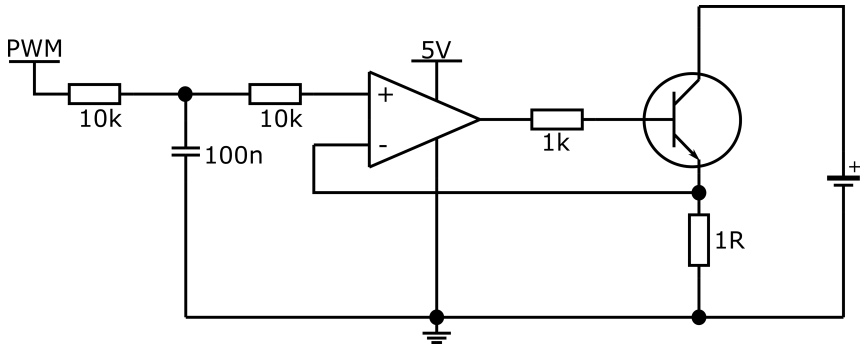
Micro controller funguje jako rozhraní mezi programem v PC a vybíjecím obvodem. K tomu, aby zvládl plnit tuto funkci, musí obsahovat několik periferií. První periferií je čítač s funkcí PWM výstupu, který slouží k řízení vybíjecího obvodu. Další periferií je AD převodník, který měří aktuální napětí baterie během vybíjecího procesu. Poslední periferií je sériová komunikace, která slouží ke komunikaci s PC programem. Ve výsledném řešení bude micro controller přijímat informaci o zvoleném vybíjecím proudu a bude odesílat aktuální naměřené napětí článku.

5.1.3 Aplikace

Ovládání vybíjecího procesu musí být realizováno aplikací s grafickým uživatelským rozhraním, tak aby bylo její použití srozumitelné a snadné pro uživatele. Aplikace bude zaznamenávat data předané ze sériové komunikace z Arduina a vhodně je vyhodnocovat a reprezentovat. Aplikace musí být schopná data vhodně uložit a uložená data bude umět i znovu načíst.

5.2 Obvod pro vybíjení baterií

Obvod na obrázku 5.1 plní tři funkce, první je filtrování PWM signálu z controleru, druhou je regulace vybíjecího proudu a poslední částí je výkonová část, která vybijí baterii. Vyfiltrovaný PWM signál slouží jako referenční signál pro regulaci proudu zajištěnou operačním zesilovačem. Jako zpětná vazba v této regulační smyčce je využit úbytek napětí na odporu, který odpovídá proudu baterie. Takto zapojený operační zesilovač se poté stará o otevření tranzistoru na požadovanou mez.



Obrázek 5.1: Původní návrh vybíjecího obvodu

5.3 Micro Controller

Jako řídící jednotka vybíječe byl vybrán procesor Atmel Atmega 2560. Tento procesor splňuje všechny podmínky shrnuté v kapitole 5.1.2. Pro usnadnění stavby funkčního prototypu byla využita deska arduino mega2560. Využitím této desky bylo vyřešeno napájení zařízení, připojení sériové komunikace přes port USB a připojení řídící jednotky ke zbytku vybíjecího obvodu.

5.4 Aplikace

Aplikaci pro řízení měření, zobrazení a správu dat je možné rozložit na několik základních částí.

- Komunikace a řízení
- Zobrazení dat
- Uložení a načtení dat
- Grafické uživatelské rozhraní

5.4.1 Komunikace a řízení

Pro komunikaci se zařízením bude použita už existující knihovna. Vzhledem k rozšířenosti Arduina je nepravděpodobné, že by taková knihovna neexistovala a tudíž nemá smysl vymýšlet vlastní implementaci komunikace.

Řízení obvodu může probíhat například posláním číselné hodnoty z aplikace na Arduino. Při odeslání nenulové hodnoty Arduino začne měřit a odesílat data. Při odeslání nuly Arduino měření a odesílání dat ukončí. Alternativou pro řízení vybíjení by mohlo být zasílání řetězcových zpráv. Pro takové zprávy by bylo nutné vymyslet vhodný formát.

5.4.2 Zobrazení dat

Bude potřeba zobrazit data, které budou přijata od Arduina. Pro jejich jednoduché a srozumitelné zobrazení se jako nejvhodnější jeví použít spojnicový graf, kde na ose x bude zaznamenán čas přijetí dat od začátku měření a na ose y bude zaznamenána naměřená hodnota napětí. Další možností by bylo například zaznamenávat data do tabulky. Toto řešení by ovšem neumožnilo snadné porovnání měření. Během měření bude nutné průběžně počítat hodnotu kapacity článku. Jelikož se bude zaznamenávat pouze poslední hodnota, tak pro její zobrazení by mohla postačit tabulka, která zobrazí hodnotu kapacity článku pro každé měření.

5.4.3 Uložení a načtení dat

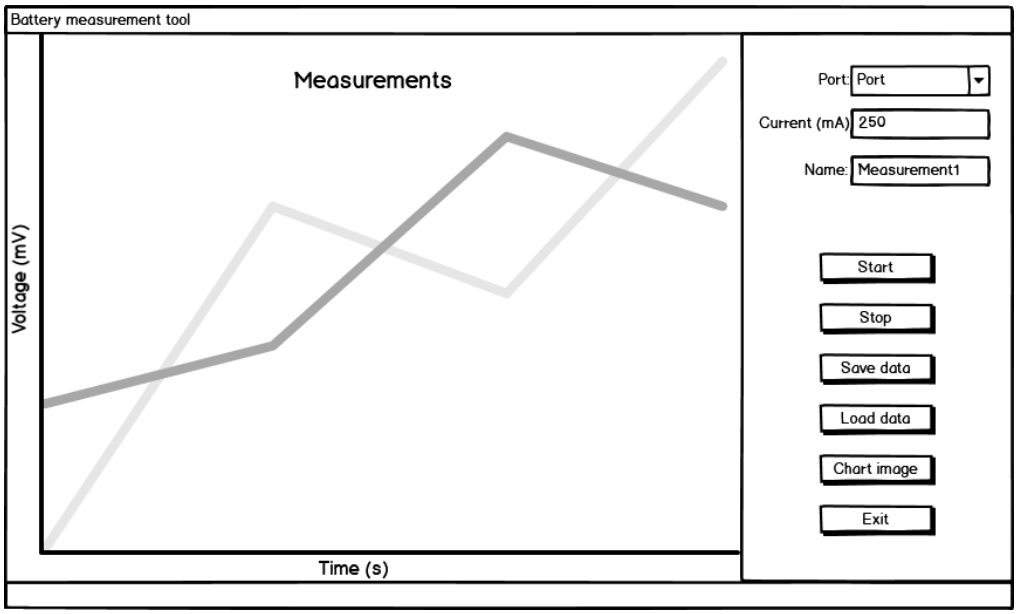
Aplikace by měla umožnit uživateli vhodným způsobem uložit a načíst data, která aplikace obdrží a zaznamená během komunikace s Arduinem. Pokud budou data zaznamenána do grafu, bylo by vhodné umožnit uživateli uložit obrázek grafu. K tomu může být použita knihovna pro grafické uživatelské rozhraní, která zároveň podporuje práci s grafy. Také by bylo vhodné umožnit uložit naměřené hodnoty napětí a času zaznamenání, aby bylo později možné je zpět do grafu načíst. Pro uložení hodnot může sloužit soubor nebo případně databáze.

Data v souboru by bylo možné uložit v textovém formátu například tak, že na první řádek by se za sebe zapsaly dvojice hodnot napětí a čas. Na další řádek by pak byla zaznamenána poslední vypočtená hodnota kapacity baterie.

Data v databázi by bylo možné uložit pomocí dvou tabulek. Jedna tabulka by sloužila pro uložení měření a jeho základních informací. Druhá tabulka by obsahovala naměřené hodnoty pro měření. S měřením by byly propojené pomocí cizího klíče, který by byl odkazoval na id měření.

5.4.4 Návrh uživatelského rozhraní

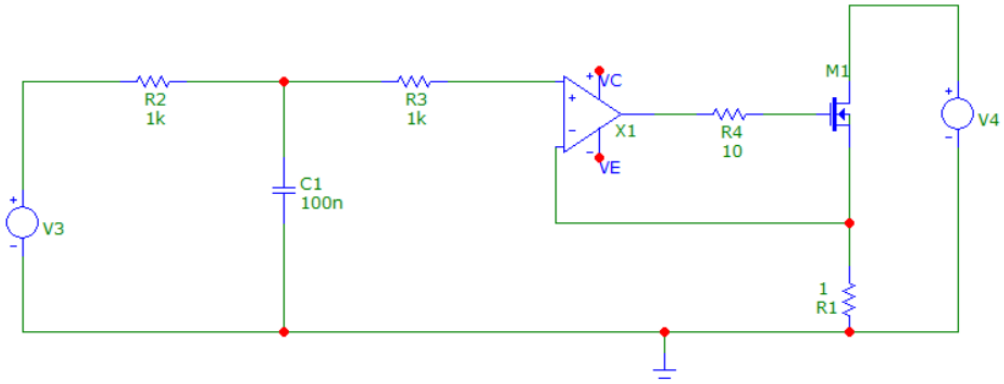
Prvotní návrh grafického uživatelského rozhraní je zobrazen na obrázku 5.2. Tento návrh byl vytvořen v programu *Balsamiq Wireframes* [17] a slouží jako předloha pro budoucí tvorbu aplikace.



Obrázek 5.2: Prvotní návrh grafického uživatelského rozhraní

6 Hardware vybíječe

Návrh hardwaru začal vytvořením simulace v programu *MicroCap* [18], pro ověření principiálního fungování navrženého obvodu. Na obrázku 6.1 lze vidět schéma zapojení v tomto programu.

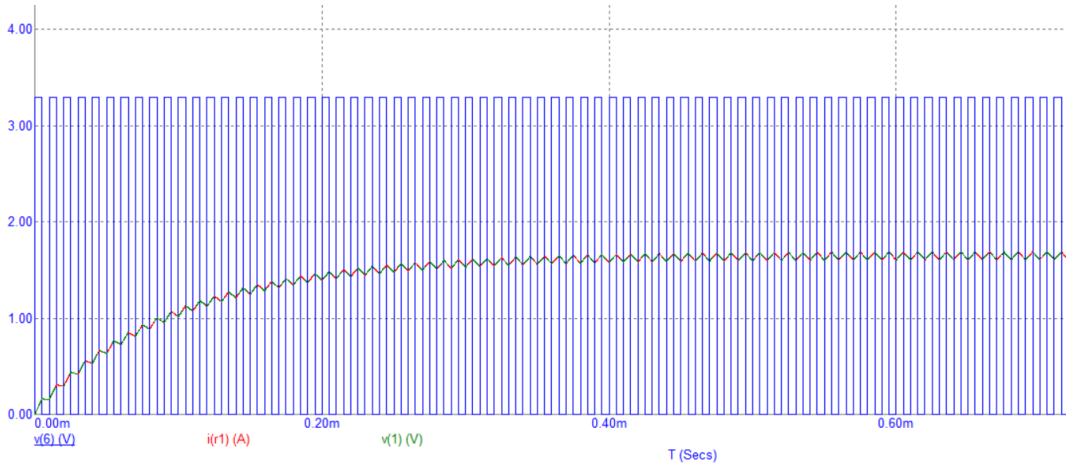


Obrázek 6.1: Navržený obvod v programu MicroCap

Zdroj V3 ve schématu reprezentuje PWM výstup mikrokontroléru. Zdroj V3 generuje obdélníkový signál, jehož střídu lze měnit a tím řídit proud odebíraný z baterie. Střída nebo také poměrné sepnutí se pohybuje v rozmezí od 0 do 1 a odpovídá poměru času, kdy je na výstup připojeno plné napětí vůči celkovému času periody obdélníkového signálu. Odpory R2, R3 a kondenzátor C1 tvoří filtr typu dolní propust, který filtruje výstupní obdélníkový signál a vytváří tak konstantní řídící napětí, které je přivedeno na operační zesilovač. Operační zesilovač se stará o buzení (otevření) tranzistoru M1 a udržuje tak napěťový úbytek na odporu R1 shodný s řídícím napětím. Úbytek napětí na odporu R1 je přímo úměrný proudu odebíranému z baterie, viz rovnice (6.1):

$$U_{R1} = I_{BAT} \cdot R_1 \tag{6.1}$$

Výsledkem je, že operační zesilovač funguje jako proporční regulátor vybíjecího proudu. Simulace viz obr. 6.2 dokazuje, že chování obvodu odpovídá předpokladům. Na průběhu je zobrazeno modře napětí PWM výstupu před filtrací. Zeleně je filtrované řídící napětí na vstupu operačního zesilovače. Červeně je zobrazen vybíjecí proud baterie, který, jak je vidět, kopíruje požadavek nastavený řídícím napětím.



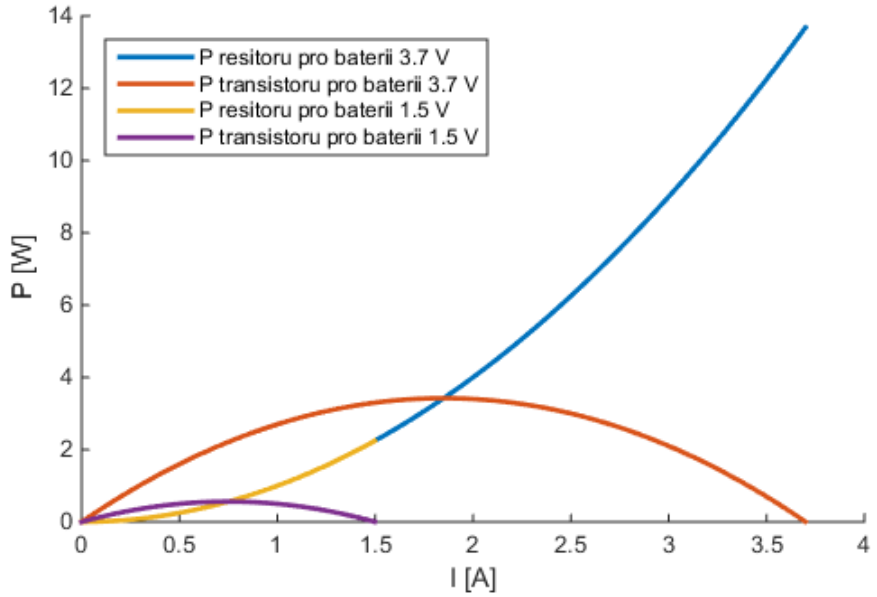
Obrázek 6.2: Výsledek simulace v programu MicroCap (Modře - PWM signál, Zeleně - filtrované řídící napětí, Červeně - vybíjecí proud)

6.1 Dimenzování

Hlavním předmětem dimenzování elektrických součástek je volba vhodných součástek, tak aby zvládly nároky, které na ně klade daná aplikace. Při dimenzování je nutné brát ohled na napětí, proud a jejich tepelné účinky na součástku. Většina těchto parametrů je pevně daná zvolenou součástkou, ale například v případě tranzistoru lze jeho pracovní rozsah upravit přidáním chladiče.

Oproti původnímu návrhu popsanému v části „Analýza“ byl výkonový obvod částečně upraven. Tyto úpravy byly nutné hlavně z hlediska výkonového dimenzování. Princip vybíječe spočívá v tom, že odebírá elektrickou energii z baterie a přeměňuje ji na teplo. Toto teplo se s ohledem na pracovní bod (nastavení vybíjecího proudu) dělí mezi tranzistor a rezistor připojený na elektrodu source na tranzistoru. Tepelný výkon na součástkách obvodu získáme jako součin proudu procházejícího součástkou a napěťového úbytku na ní.

Jelikož je zařízení určeno pro vybíjení článků s různým napětím, kde proud je proměnný a záleží na zadání velikosti proudu od uživatele, je potřeba kontrola dimenzování součástek pro různé provozní stavy. V závislosti na napětí článku a nastaveném vybíjecím proudu se mění poměr rozložení výkonu mezi odpor a tranzistor. Na obrázku 6.3 je ukázána závislost výkonu na součástkách vybíjecího obvodu v závislosti na velikosti vybíjecího proudu pro dva různé typy elektrických článků.

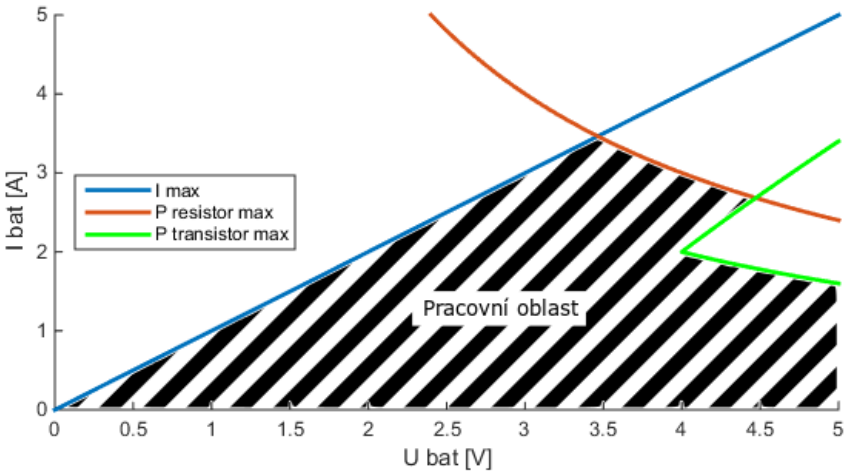


Obrázek 6.3: Závislost tepelného výkonu součástek na proudu vybíjecím obvodem

Při dimenzování rezistoru bylo postupováno od toho nejhoršího případu. Z pohledu tepelného výkonu na rezistoru je to stav, kdy obvodem prochází maximální proud. Maximální proud obvodem prochází ve chvíli, kdy je tranzistor plně sepnut. Takový stav situaci komplikuje, jelikož v tomto případě je celý výkon dodávaný do obvodu z baterie spotřebováván pouze na rezistoru. Pro dostatečný pracovní rozsah zařízení byla namísto jednoho rezistoru zvolena sériově paralelní kombinace čtyř rezistorů o celkovém výkonu 12W a odporu 1Ω.

Z pohledu tranzistoru dochází k nejhoršímu stavu v polovině pracovní charakteristiky, kdy obvodem teče poloviční proud vůči maximálnímu (danému napětím baterie). V tomto pracovním bodě je úbytek napětí na tranzistoru roven polovině napětí baterie. Z toho vyplývá, že maximální tepelný výkon tranzistoru je $\frac{1}{4}$ maximálního tepelného výkonu rezistoru.

Hledáním vhodného tranzistoru pro použití v obvodu se ukázala nevhodnost původně zvoleného bipolárního tranzistoru. Bipolární tranzistory jsou řízeny proudem do elektrody báze a zároveň bipolární tranzistory, které splňují podmínky, mají velmi nízkou hodnotu zesílení. Pro představu, zesílení takovýchto tranzistorů se pohybuje v řádu desítek, což by velmi zkomplikovalo výběr operačního zesilovače, který se stará o buzení tranzistoru. Buzení bipolárního tranzistoru by vyžadovalo proudy v řádu stovek miliampér, což by například aktuálně použitý zesilovač nezvládl, jelikož má výstupní proud maximálně 25mA. Maximální výstupní proud 25mA není v případě operačních zesilovačů nijak nezvykle nízký, jelikož se jedná o zařízení určená ke zpracování a zesílení signálů, vysoké výkony se na jejich výstupu neočekávají. Zvolený operační zesilovač měl výhodu v tom, že se jednalo o nízko-nákladové zařízení se schopností práce rail-to-rail, což umožnilo využít 5V napájení procesoru i pro operační zesilovač. Problém s omezením proudu pro buzení tranzistoru byl vyřešen volbou tranzistoru typu *MOSFET* (Metal Oxide Semiconductor Field Effect Transistor), konkrétně jeho variantou s kanálem typu N, která je řízena napětím mezi elektrodou gate a source. Tento unipolární tranzistor je řízen napětím. Proud do elektrody gate je v ustáleném stavu nulový, proto neklade tak velké nároky na budící obvod (operační zesilovač).



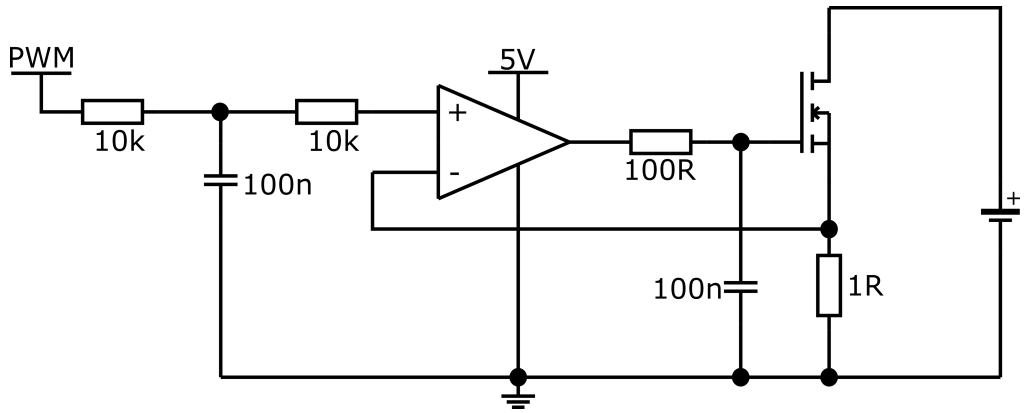
Obrázek 6.4: Pracovní oblast zařízení

Použitý tranzistor má v katalogu deklarovaný tepelný odpor mezi čipem a okolím 62 K/W. Bez chladiče by tedy nejvyšší dovolený výkon dosahoval pouze hodnoty kolem 1,6 W, proto musel být doplněn o chladič, který tepelný odpor vůči prostředí snížil přibližně na 25K/W, a tím zvýšil maximální tepelný výkon tranzistoru na přibližně 4W.

Výsledná pracovní oblast daná výkonovými omezeními a maximální možným vybíjecím proudem je na obrázku 6.4.

6.2 Testování funkčnosti zařízení

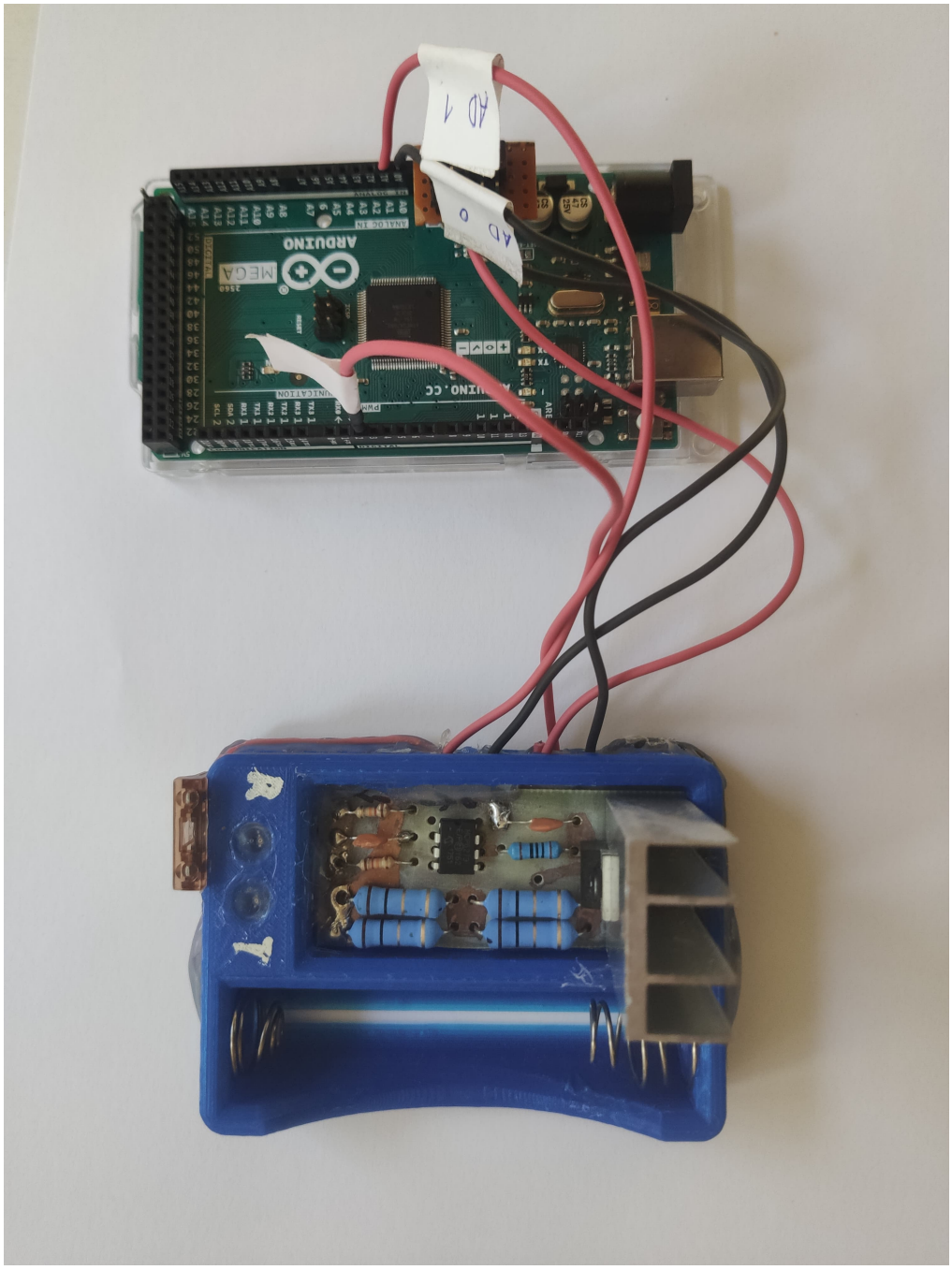
Před konečným sestavením vybíjecího obvodu byl ještě proveden finální test reálného zařízení zapojením obvodu na nepájivém poli. Test ukázal nestabilní chování způsobené vysokým zesílením a krátkou časovou konstantou unipolárního tranzistoru, proto byl obvod doplněn o kondenzátor připojený na elektrodu gate. Tento kondenzátor zpomalil odezvu výkonového obvodu a tím systém stabilizoval. Pro výsledné schéma včetně hodnot pasivních prvků viz obr. 6.5.



Obrázek 6.5: Schéma vybíjecího obvodu

6.2.1 Stavba vybíjecího obvodu

Pro usnadnění dalšího vývoje softwaru a samotných testů baterií byl dle schématu vytvořen plošný obvod. Protože se jedná o poměrně jednoduchý obvod, byla zvolena technika ručního kreslení spojů a leptání v persíranu sodném. Výsledný obvod byl poté vlepen do plastového rámečku. Rámeček je vyroben pomocí 3D tisku a jeho součástí je i držák baterie s pružinovými kontakty. Skutečná podoba výsledného obvodu propojeného s Arduinem je na obrázku 6.6.



Obrázek 6.6: Schéma vybíjecího obvodu

7 Software vybíječe

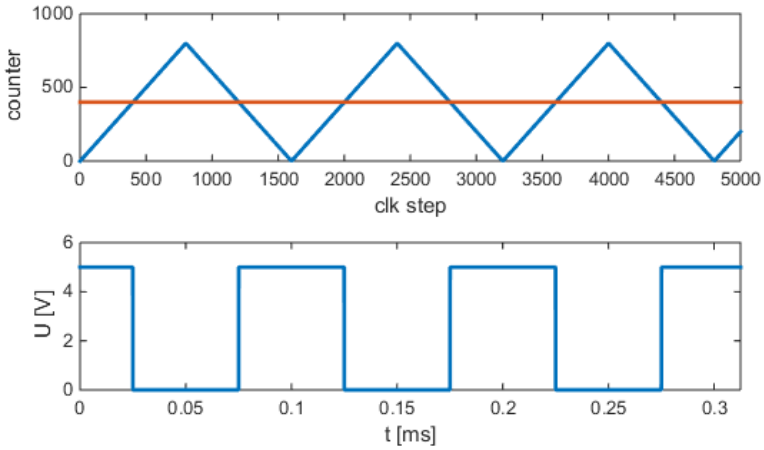
Řídicí jednotka vybíječe je tvořena pomocí *MCU Atmel ATmega 2560* osazeném na desce *Arduino MEGA*. Při vývoji nebyly využity žádné knihovny určené pro jednoduché programování Arduina, Arduino bylo využito jen z důvodu usnadnění realizace hardwaru. Pro vývoj softwaru bylo využito *Atmel studio 7.0*.

7.1 Nastavení PWM

Pro možnost řízení vybíječe byl využit *PWM* (Pulse Width Modulation) výstup. Základem modulátoru je jeden z vnitřních čítačů MCU. Z možných módů nastavení čítače byl vybrán fázově a frekvenčně korektní chod. Toto nastavení znamená, že čítač vytváří nosný signál (pilu) tím, že načítá pulzy zvoleného časového signálu až do předem určené hodnoty (perioda nosného signálu), kdy začne čítač odečítat zpět do nuly. Výsledkem je, že ve vnitřním registru čítače vznikne symetrický pilový signál, viz obr. 7.1. Dalším specifikem tohoto nastavení je, že k aktualizaci komparační hodnoty dochází vždy v nule pilového signálu. K tomu jsou využity takzvané shadow registry, do kterých se požadovaná hodnota zapíše okamžitě, ale do čítače se nahraje až při splnění nějaké podmínky. V tomto případě dosažení spodního bodu pily.

Čítač se dále stará o komparaci nosného signálu s modulačním, který odpovídá požadavku na střídu výstupního signálu. Modulační signál nabývá hodnot od 0 do velikosti periody nosného signálu, čemuž potom odpovídá velikost střídy výstupního signálu od 0 do 1. Výstupem z čítače (PWM modulátoru) je tedy obdélníkový signál. Nastavitelnými parametry tohoto signálu jsou jeho frekvence a střída. Frekvence (f_{out}) odpovídá frekvenci hodin čítače (f_{clk}) a registru udávajícímu počet tiků na periodu nosného signálu (ICRn). Střída, jak už bylo zmíněno, odpovídá velikosti modulačního signálu (OCRn). Díky doplnění PWM výstupu o filtr typu dolní propust popsaném v kapitole Hardware vybíječe odpovídá střední hodnota výstupního signálu velikosti vybíjecího proudu baterie. Střední hodnota (U_{av}) se dá spočítat jako součin amplitudy (U_{amp}) a střídy výstupního obdélníkového signálu.

$$f_{out} = \frac{f_{clk}}{2 \cdot ICRn}[Hz] \tag{7.1}$$



Obrázek 7.1: Průběh nosného a modulačního signálu (nahoře), průběh odpovídajícího výstupního napětí (dole)

$$U_{av} = U_{amp}\frac{OCRn}{ICRn}[V] \tag{7.2}$$

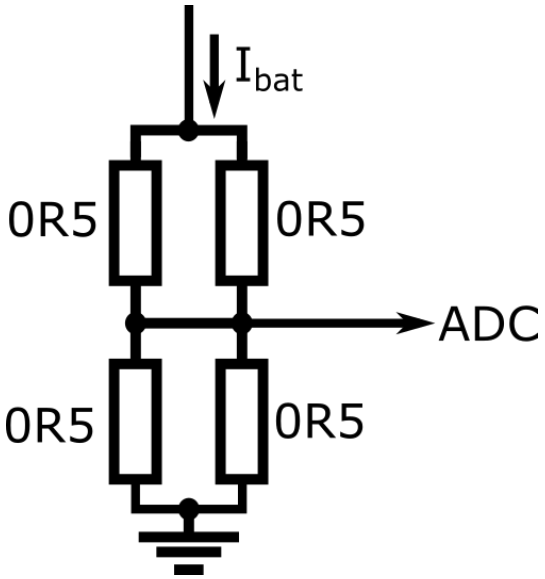
7.2 Nastavení AD převodníku

Pro měření aktuálního stavu vybíjecího procesu jsou využity AD převodníky, které jsou součástí MCU. MCU nabízí až 16 měřících kanálů. Některé kombinace měřících kanálů lze využít k diferenciálnímu měření. V případě čtyř kombinací pro diferenciální měření je možné měření doplnit až 200 násobným zesílením. Doba jednoho převodu se pohybuje od 13 do 260μs v závislosti na zvolené přesnosti. Převod je založen na porovnání měřeného napětí s referenčním.

Jelikož bude AD převodník použit v single conversion módu, bude převodník vyžadovat pokyn k zahájení měření. K tomu je využit čítač, který každých 100 ms spustí měření. Po konci měření nastane přerušení, kde je do příslušné proměnné načtena změřená hodnot a provedeny další operace. Jelikož chceme měřit proud i napětí baterie, je nutné při měření přepínat mezi dvěma kanály. Toto přepnutí je realizováno během právě zmíněného přerušení. Pomocí podmínky if je zjištěno aktuální nastavení kontrolního registru AD převodníku. Podle toho, jaký kanál byl právě aktivní, je výsledek měření načten do příslušné proměnné a měření je přepnuto na další kanál.

7.2.1 Měření proudu

Proud baterie je měřen jako úbytek napětí na odporu. Odpor, který je k tomuto účelu využit, současně také slouží pro nastavení pracovního bodu tranzistoru. Měření vychází z Ohmova zákona. Vzhledem k tomu, že je měření připojeno doprostřed sestavy odporů, je měřící odpor $R = 0,5\Omega$ a z toho plyne $I_{bat} = 2 \cdot U_R$. Zapojení měření do sestavy odporů je ukázáno na obrázku 10.1. Do výpočtu měřící konstanty je pak potřeba zohlednit ještě několik proměnných. Rozlišení AD převodníku, které je 10 bitů, což znamená, že maximální hodnota výsledku převodu je 1023. Poslední důležitou informací je referenční napětí, ke kterému je měření vztaženo. Jsou tři hlavní možnosti, jak toto referenční napětí zvolit. První možností volby referenčního napětí je napájecí napětí MCU. Tato volba může způsobovat nepřesnosti v měření, neboť napájecí napětí může mírně kolísat v závislosti na zatížení, nebo aktuálně použitém napájecím zdroji. Další možností je využití externího referenčního napětí připojeného na příslušný pin MCU. Tato možnost zajistí přesnější měření, ale vyžaduje doplnění MCU o vhodný obvod. Poslední možností je využití interních referenčních napětí 1,1 nebo 2,56V. Pro naše měření je zvoleno interní referenční napětí 2,56V, což umožňuje měřit proudy od 0 do 5,12A. Konstanta měření je pak ještě upravena tak, aby výsledná hodnota byla v mA.

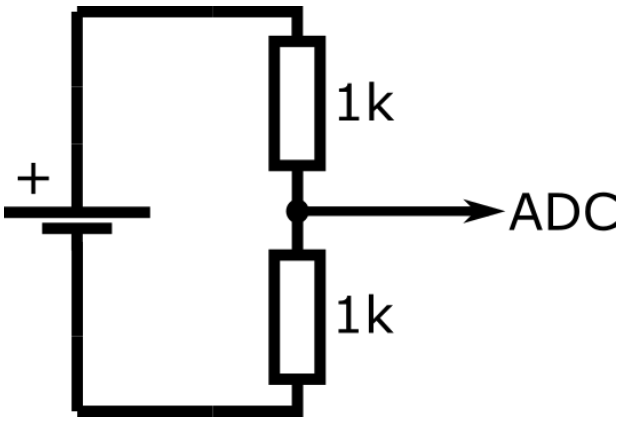


Obrázek 7.2: Schéma zapojení měření proudu baterie

$$I_{bat} = ADC \cdot \frac{5,12 \cdot 1000}{1023} = ADC \cdot 5[mA] \tag{7.3}$$

7.2.2 Měření napětí

Měření napětí je provedené podobně jako měření proudu. Měřící obvod je doplněn o napěťový dělič, který snižuje napětí baterie na polovinu (viz obr. 7.3). Díky tomuto děliči je možné měřit i napětí větší, než je hodnota referenčního napětí.



Obrázek 7.3: Schéma zapojení napěťového děliče pro měření napětí baterie

Napěťový dělič zvýší rozsah měření napětí článku z 2,56V na 5,12V. Konstanta měření je opět upravena tak, aby změřené napětí bylo v mV.

$$U_{bat} = ADC \cdot \frac{5,12 \cdot 1000}{1023} = ADC \cdot 5[mV] \tag{7.4}$$

7.2.3 Sériová komunikace

Sériová komunikace je nastavena tak, aby bylo Arduino schopno spolupracovat s programem v PC. Prvním krokem je tedy nutnost shodně nastavit parametry sériové komunikace. Mezi ně patří například frekvence, počet bitů na zprávu nebo paritní bity. Dále bylo nutné vytvořit kód, který bude správným způsobem reagovat na přijatá data a odesílat změřená data.

Na začátku měření je z PC odeslána zpráva, která odpovídá nastavenému vybíjecímu proudu. Tato zpráva je signálem pro Arduino, aby započalo vybíjení baterie zvoleným proudem. Obdobně je vybíjení baterie ukončeno přijetím požadavku na nulový proud.

Způsob odesílání dat souvisí s tím, jakým způsobem jsou data zpracována v PC programu, což je popsáno v kapitole Průběh Měření. Změřená desetibitová hodnota tedy musí být před odesláním rozdělena tak, aby bylo možné jí poslat ve dvou zprávách, jejichž velikost je omezena na 8 bitů.

8 Aplikace pro řízení vybíjení

Aplikace poskytující uživatelské rozhraní uživateli a zobrazení dat je napsaná v jazyce Java 11. K vytvoření uživatelského rozhraní byla použita knihovna *JavaFX*, která je považována za standard při vytváření grafických uživatelských rozhraní pro Java aplikace.

8.1 Použité knihovny a nástroje

Při vývoji aplikace pro řízení vybíjení článku a zobrazování dat bylo využito několik existujících knihoven a nástrojů. Tyto knihovny a nástroje slouží zejména ke správě projektu a jeho závislostí, vytváření grafického uživatelského rozhraní, atd. Výčet všech použitých knihoven a nástrojů včetně jejich stručného popisu je uveden dále.

8.1.1 Apache Maven

Maven je nástroj pro správu a sestavování aplikací. Podporovaný je především jazykem Java. Informace potřebné ke kompilaci a sestavení jsou uloženy v souboru pom.xml (project object model) nacházející se v kořenovém adresáři každého projektu. Jsou zde nadefinované závislosti na externích knihovnách. Maven tyto knihovny pak automaticky dohledá a nainstaluje. [19]

8.1.2 JavaFX

JavaFX je framework sloužící ke tvorbě aplikací s grafickým uživatelským rozhraním. Podporuje grafy, tabulky, obrázky a další technologie. [20]

8.1.3 jSerialComm

Tato knihovna pro Javu je určená k poskytování přístupu na standardní sériové porty nezávislého na platformě bez nutnosti externích knihoven, nativního kódu, nebo jiných nástrojů. Lze díky ní jednoduše vyčíst porty připojené k počítači, nastavit timeouty portů pro čtení i zápis nebo posílat a přijímat toky bajtů přes rozhraní Java InputStream a OutputStream. [21]

8.1.4 Jackson

Knihovna Jackson [22] je použita pouze pro mapování Java objektů na JSON řetězce (JavaScript Object Notation) a naopak, což je odlehčený formát pro výměnu dat. [23]

8.2 Struktura programu

Zdrojový kód programu je rozdělen do čtyř základních balíků a to jsou *app*, *gui*, *communication* a *data*. V balíku app se nachází hlavní třída programu. Dále zde jsou třídy, které obstarávají například zpracování a ukládání dat. V balíku *gui* se nachází pouze jedna třída, která definuje uživatelské rozhraní. Aplikace se skládá pouze z jednoho okna, proto zde není více tříd. Třída v balíku communication slouží jako vlákno pro komunikaci s Arduinem. Balík data obsahuje pouze třídy, které reprezentují data v grafu a data o měření. Slouží zejména k ukládání a načítání dat ze souborů.

8.2.1 BatteryMeasuringTool

Hlavní třída programu BatteryMeasuringTool se nachází v balíku app a slouží pouze ke spuštění třídy, která inicializuje uživatelské rozhraní.

Od Javy verze 11 už není knihovna JavaFX součástí JDK. Z toho důvodu byl použit nástroj Maven, pro správu projektu a závislostí. Bohužel v aplikaci napsané v Javě 11 a při použití knihovny JavaFX za použití nástroje Maven nelze použít hlavní třídu programu zároveň jako třídu inicializující grafické uživatelské rozhraní, protože by se aplikace nespustila. Z toho důvodu musí být hlavní třída oddělena.

8.2.2 DataHandler

Třída se nachází v balíku app a slouží k ukládání obrazů grafu na disk a zároveň k ukládání a načítání dat z grafu. Tato třída je implementována jako *singleton* [24] a je používána pouze prostřednictvím třídy MeasurementWindow. Používá se při akcích uživatele pro uložení nebo nahrání dat.

8.2.3 MeasurementManager

MeasurementManager je třída z balíku app. Slouží pro zahajování a ukončování komunikace mezi aplikací a hardwarovým zařízením vybíjejícím baterii. Tato třída je také implementována jako singleton a je také používána pouze prostřednictvím třídy MeasurementWindow. Používá se při akcích uživatele,

které zahajují a ukončují měření. MeasurementManager vždy vytváří pouze jedno vlákno pro komunikaci

8.2.4 Utility

Třída Utility je v balíku app a obsahuje podpůrné metody například pro převod času do požadovaného formátu nebo rozšíření bajtového pole. Dále také načítá řetězcové zdroje, které jsou použity pro konstantní řetězce použité v oknech a dialozích aplikace.

8.2.5 MeasurementWindow

Tato třída nacházející se v balíku gui zajišťuje grafické uživatelské rozhraní, které zobrazuje data z měření a umožňuje uživateli ovládat aplikaci. Používá třídy MeasurementManager a DataHandler, jejichž metody využívá při akcích uživatele, jako je například zahájení měření, ukončení měření, uložení dat na do souboru na disk a načtení dat ze souboru na disku do aplikace.

8.2.6 ArduinoCommunicationThread

Třída slouží jako vlákno, které spouští a ukončuje třída MeasurementManager. Nachází se v balíku communication. Získává data z Arduina a ukládá je přímo do grafu. Dále také průběžně po každé přijaté hodnotě vypočte hodnotu kapacity baterie a zaznamená ji do tabulky kapacit měření. Aplikace neumožňuje, aby do grafu zapisovalo data více vláken současně.

8.2.7 ChartData

Třída se nachází v balíku data. Slouží pouze pro reprezentaci naměřených dat. Třída Measurement obsahuje pole těchto tříd, které tedy ve výsledku představuje celé měření.

8.2.8 Measurement

Tato třída obsahuje data z celého měření a poslední vypočtenou hodnotu kapacity. Zároveň slouží pro ukládání dat na disk počítače a jejich opětovné načítání dat zpět do aplikace. Třída se nachází v balíku data.

8.3 Implementace

Aplikace během svého běhu na základě akcí uživatele vykonává několik základních úloh. Mezi tyto úlohy patří navázání spojení s Arduinem a zahájení měření, zaznamenávání hodnot z měření vybíjení článku a jejich vykreslování v grafu, ukládání naměřených dat a obrázků grafu a načítání dříve provedených měření do grafu. V této kapitole je popsáno, jak tyto úlohy pracují. Obrázek výsledného GUI je v uživatelské příručce v příloze.

8.3.1 Zahájení měření

Po spuštění aplikace se uživateli zobrazí grafické uživatelské rozhraní. Následně uživatel musí vybrat port, přes který je k počítači připojeno zařízení pro vybíjení baterií, musí zadat hodnotu vybíjecího proudu baterie a jako poslední musí zadat název měření. Po spuštění měření aplikace nejprve zhodnotí validitu zadaných údajů, tzn. zkontroluje, že byl zadán port a že zadaný název měření vyhovuje omezení na 1 až 16 alfanumerických znaků s podtržítkem. Pokud jsou zadané údaje v pořádku, aplikace vytvoří nové vlákno pro komunikaci s Arduinem. Vlákno se nejprve pokusí otevřít port vybraný uživatelem, a pokud otevření portu proběhne úspěšně, tak vlákno odešle uživatelem zvolenou hodnotu proudu Arduinu, pro které je tato hodnota znamením, že má začít odesílat data a spustit tak vybíjení baterie.

8.3.2 Průběh měření

Pro přijímání dat od Arduina je použit blokující socket, který má nastavený timeout na jednu minutu, tzn. vlákno čeká na instrukci čtení dat z portu a pokud po dobu jedné minuty nepřijdou od Arduina žádná data, tak se spojení ukončí. Tento limit je nastaven, aby nedošlo k situaci, že vlákno přestane přijímat data a uživatel by neměl možnost jej ukončit. Arduino odesílá data jako jednotlivé bajty, pro reprezentaci dat, které od Arduina aplikace přijímá, stačí dva bajty. Tyto hodnoty se zapíší do bajtového pole o velikosti dva, které se následně převede na integer a aby bylo možné toto bajtové pole převést na integer, tak je nejprve nutné jej rozšířit na bajtové pole o velikosti čtyři. Dva bajty totiž nestačí k reprezentaci integeru, který má v jazyce Java čtyři bajty. Data se do grafu se zapisují jako dvě celá čísla, kde první číslo je hodnota obdržená od Arduina a druhé číslo jsou vteřiny od začátku měření. Tento čas si počítá vlákno samo. Pseudo kód metody vlákna pro komunikaci s arduinem je zobrazen v algoritmu 8.1

Algoritmus 8.1: Algoritmus pro komunikaci s arduinem

```
port -> Port to which hardware device is connected
stopFlag -> Determines whether thread should continue
            running
chart -> Chart in which data is recorded
capacityMap -> Map in which capacity value is recorded
               for each measurement
measurementName -> Name of the measurement chosen by user
current -> current used for battery discharge

procedure run()

  port.openPort()
  port.setBlocking()
  port.setTimeout(60000) // timeout in milliseconds
  port.send(current)

  while !stopFlag do
    currentTimeFromStart // calculated time from
                            the thread start
    byte[2] bytes
    port.read(bytes, 2)
    bytes = expandTo4Bytes(bytes)

    if !stopFlag then

      chart.add(currentTimeFromStart, toInt(bytes))
      capacityMap.put(measurementName,
          calculateCapacityValue(currentTimeFromStart,
              current
          )
      )
    end if
  end while
end procedure
```

Vlákno průběžně počítá kapacitu, po každém přijetí hodnot od Arduina. Jak již bylo zmíněno v teoretické části, kapacita baterie se udává, jako integrál z proudu baterie, podle času. Jelikož se v našem případě velikost vybíjecího proudu nemění, lze nahradit při výpočtu integrál prostým součinem. Kapacita baterie Q_{bat} v mAh pak odpovídá součinu proudu baterie I_{bat} v mA a obě vybíjecího procesu t v hodinách.

$$Q_{bat} = I_{bat} \cdot t[mAh; mA, h] \tag{8.1}$$

8.3.3 Ukončení měření

Při ukončení měření se přes socket na arduino odešle hodnota 0. Tato hodnota signalizuje, že arduino má ukončit odesílání hodnot a následně se ukončí spojení s portem. Ukončení měření je podmíněno akcí uživatele, který buď zastaví měření, vymaže obsah grafu nebo ukončí celou aplikaci. Vlákno při běhu používá hodnotu typu boolean k určení toho, jestli má v běhu pokračovat nebo skončit. Při ukončení vlákna tedy dojde k tomu, že tato proměnná se nastaví na hodnotu true, čímž ukončí nekonečný cyklus v metodě vlákna. K ukončení vlákna tedy nemusí dojít okamžitě, protože metoda vlákna může být v tu chvíli zastavená na instrukci čtení dat ze socketu a v tom případě ihned nereaguje na pokyn k zastavení, ale ukončí se až když dojde zpět na začátek metody a neprojde přes podmínku cyklu while. Nicméně přesto, že vlákno se neukončí okamžitě, už nemůže přistupovat ke grafu a zapisovat data.

8.3.4 Ukládání dat

Aplikace umožňuje dva způsoby uložení dat. Prvním způsobem je uložení obrázku grafu ve formátu .png. Název obrázku si uživatel volí sám, ale výsledný obrázek se vždy ukládá do složky charts, která se nachází v adresáři projektu. Tato složka se automaticky vytvoří až při první potřebě uložit obrázek grafu. Aplikace neumožňuje uložit prázdný graf. Postup pro uložení grafu je popsán v algoritmu 8.2.

Algoritmus 8.2: Algoritmus pro uložení grafu

```
1  chartsDir -> Directory where chart images are stored
2  chart -> chart that will be saved
3
4  chartName -> name specified by user
5
6  procedure exportChart(chart, chartName)
7      fileExtension = ".png"
8
9      chartFile = create new File(chartsdir, chartName +
10                             fileExtension)
11
12     chartFile.mkDirs()
13     writeToFile(chart.getImage(), chartFile)
14 end procedure
```

Druhý způsob je uložení dat měření z grafu na disk. Při zvolení této možnosti se vždy uloží veškerá měření, která jsou v grafu zobrazená. Pro každé měření se vytvoří samostatný soubor, který má stejné jméno jako dané měření a navíc obsahuje informaci jak velký proud byl použit k vybíjení baterie. Data jsou ukládány do souborů s příponou .json. Obsah souboru má textový formát a obsah je *json* reprezentací objektu, který obsahuje všechny hodnoty daného měření a poslední vypočtenou hodnotu kapacity. Soubory se ukládají do složky data, která se nachází v adresáři projektu. Postup pro uložení naměřených dat na disk je popsán v algoritmu 8.3.

Algoritmus 8.3: Algoritmus pro uložení naměřených dat

```
1  chartData[] -> Array of data for each measurement
2  dataDir -> Directory where measurements data is stored
3
4  procedure exportData(chartData[])
5      fileExtension = ".json"
6
7      for each data in chartData do
8          dataStorage = create new DataStorage(
9                            copyValuesFromData(),
10                           getCapacityFromMap)
11
12         jsonMapper = create new JsonMapper()
13         dataJson = jsonMapper.getObjectAsJson(
14                                 dataStorage)
15
16         dataFile = create new File(dataDir,
17                                    data.getName() +
18                                    fileExtension)
19
20         writeToFile(dataJson, dataFile)
21
22     end for
23 end procedure
```

8.3.5 Načítání dat

Data se načítají vždy po jednom souboru. Název souboru se v aplikaci použije jako název měření, přičemž není možné do aplikace přidat dvě měření se stejným názvem. Název souboru se skládá ze dvou částí, které jsou od sebe odděleny pomlčkou. Část názvu před pomlčkou odpovídá názvu měření a část názvu za pomlčkou odpovídá hodnotě v miliampérech. Obsah souboru se celý přečte a následně se pomocí knihovny *Jackson* převede

na objekt, který reprezentuje celé měření a poslední vypočtenou hodnotu kapacity. Data jsou následně zanesena do grafu a tabulky. Postup načítání dat je popsán v algritmu 8.4.

Algoritmus 8.4: Algoritmus pro načtení uložených dat z měření

```
1  file -> file picked by user from file selection dialog
2  chart -> chart where data will be saved
3  capacityMap -> map of capacity for each measurement
4
5  procedure importData(file)
6
7      dataJson = readFromFile(file)
8      jsonMapper = create new JsonMapper()
9      dataStorage = jsonMapper.readJsonAsObject(dataJson,
       DataStorage.class);
10     dataName = file.getName()
11     chartData = copyValuesFromStorage(dataStorage)
12     chartData.setName(dataName)
13
14     if capacityMap.containsKey(dataName) then
15         return null;
16     else
17         capacityMap.add(dataName, data.getCapacity());
18         return series;
19     end if
20 end procedure
```

9 Omezení a možnosti rozšíření

Jedním z velkých omezení vytvořeného řešení je, že nedokáže měřit z více zařízení najednou. Samozřejmě je možné vytvořit více hardwarových zařízení pro vybíjení baterií, ale software pro ovládání arduina ani aplikace nejsou vytvořené tak, aby byly schopny vybíjet a zaznamenávat data z několika baterií současně. Při použití dvou Arduin, které by každé bylo připojené k PC v samostatném USB portu, by se neschopnost řídit vybíjení a zaznamenávat data z více baterií přenesla pouze na aplikaci.

Během vývoje zařízení byla implementována i funkce pozastavení vybíjecího procesu, ovšem později byla odstraněna. V současnosti tedy není možné pozastavit probíhající měření. Během manuálního testování se ukázalo, že tato funkce není příliš přínosná, jelikož deformuje tvar vybíjecí charakteristiky. Prvním důvodem deformace je průběh vybíjecího procesu. Na začátku měření je poslána hodnota napětí nezatížené baterie (proud baterie je nulový). Tato hodnota způsobí krátký puls na vybíjecí charakteristice. Ten by bylo možné snadno odstranit, proto je větší druhý problém. Ten je spojen s vnitřním fungováním elektrochemického článku. Během pauzy se článek částečně zregeneruje, stoupne jeho napětí a trvá poměrně dlouho, než se dostane zpět na původní průběh napětí. Ze stejných důvodů nelze pokračovat ani v nahraném měření ze souboru.

Jak lze vidět z obrázku 6.4 pracovní oblast je omezena několika kritérii. Výkonové omezení dané tranzistorem a rezistorem lze snadno upravit volbou odolnějších součástek. Větší problém v omezení pracovní oblasti je způsoben principem fungování obvodu, přesněji v maximální vybíjecím proudu označeném jako I_{max}. Tento proud, jak lze pozorovat, je přímo úměrný velikosti napětí článku. Sklon křivky I_{max} je dán velikostí odporu pro nastavení pracovního bodu, a i když je možné její sklon změnou velikosti tohoto odporu upravit, bude tato závislost stále lineární s průchodem nulou. Při požadavku na výrazné rozšíření pracovní oblasti by bylo nutné změnit topologii vybíjecího obvodu, například na spínaný obvod typu buck-boost, což je typ stejnosměrného měniče, který zvládne zvyšovat nebo snižovat vstupní napětí (napětí baterie). Tato změna by ale měla výrazný dopad na cenu odvodu a složitost regulace.

10 Závěr

Práce začíná úvodem do problematiky elektrochemických článků. Obsahuje krátký popis existujících druhů galvanických článků a přehled parametrů důležitých při výběru článku. Dále je v úvodu také seznam elektrických součástek potřebných k vytvoření vybíjecího obvodu.

Hlavním tématem práce je popis navrženého softwaru a hardwaru vybíječe primárních článků. Kromě popisu funkce je stručně popsán i postup tvorby jednotlivých částí s důrazem na kritické části návrhu.

Fyzická část vybíječe je tvořena výkonovým vybíjecím obvodem a řídícím obvodem. Výkonový obvod je dimenzován tak, že zvládne vybíjet články o napětí až 5V. Proudová zatížitelnost záleží na aktuálním napětí článku, maxima dosahuje pro článek o napětí 3,5V, který lze vybíjet proudem až 3A. O regulaci vybíjecího proudu se stará operační zesilovač, který je řízen pomocí PWM signálu z MCU.

Firmware zařízení nahraný v MCU ATmega 2560 funguje jako rozhraní mezi PC a vybíjecím obvodem. Z PC přichází impuls ke spuštění vybíjení současně s nastavením velikosti vybíjecího proudu. Zpět do PC je pak pravidelně odesílána aktuální hodnota napětí článku.

Pro řízení vybíjení baterie a zpracování dat byla v jazyce Java vytvořena aplikace s grafickým uživatelským rozhraním. Aplikace zobrazuje obdržená data a průběžně počítá hodnotu kapacity momentálně vybíjené baterie, kterou zobrazuje v tabulce. Aplikace také umožňuje uložení obrázku grafu a uložení měření do souboru, který je možné kdykoliv později nahrát zpět do aplikace.

Vytvořený obvod, software pro jeho ovládání i aplikace pro řízení vybíjení a zaznamenávání hodnot dohromady tvoří celkové řešení pro vybíjení baterií, zaznamenávání hodnot o vybíjené baterii, zobrazení zaznamenaných hodnot, výpočtu dalších hodnot a jejich uložení na pevný disk počítače.

Funkčnost konečného zařízení byla otestována změřením vybíjecích charakteristik několika článků. Během testování se neprojevily žádné chyby a výsledné charakteristiky mají očekávaný tvar. Byly změřeny AA baterie od různých výrobců a všechny měly kapacitu kolem 2,1Ah. Součástí pokusů bylo i s srovnání konzistentnosti výroby a byly změřeny charakteristiky dvou shodných baterií. Zobrazení výsledných charakteristik lze najít v příloze.

Literatura

[1] Galvanické články. URL http://konstrukt.wz.cz/encyklopedie_html/fyz2.htm.

[2] *Akumulátory od principu k praxi*. FCC PUBLIC, Praha, 2003. ISBN 80-86534-03-0.

[3] Zinko-uhlíková baterie. URL http://www.bateria.cz/stranky3/zabava--pouceni/jak-to-funguje-/zinkouhlikova-baterie.html.

[4] Primární galvanické články a jejich porovnání. *ELEKTRO*. URL http://www.odbornecasopisy.cz/elektro/casopis/tema/primarni-galvanicke-clanky-a-jejich-porovnani--13134.

[5] Alkalická baterie. URL http://www.bateria.cz/stranky3/zabava--pouceni/jak-to-funguje-/alkalicka-baterie.html.

[6] Abeceda baterií a akumulátorů. URL http://www.battex.info/.

[7] Spouštěcí akumulátory, funkce, druhy, vlastnosti, zásady údržby a péče o akumulátory. URL https://publi.cz/books/160/03.html.

[8] What is voltage in a battery? URL https://sciencing.com/voltage-battery-5058989.html.

[9] Obvody stejnosměrného proudu, . URL http://old.spsemoh.cz/vyuka/zae/el4.htm.

[10] What is internal resistance? how a battery works?, . URL https://www.scienceabc.com/innovation/internal-resistance-battery-works.html.

[11] How does rising internal resistance affect performance?, . URL https://batteryuniversity.com/learn/article/rising_internal_resistance.

[12] Operational amplifiers. URL https://www.elprocus.com/operational-amplifiers/.

[13] J.P. Vasseur. *Properties and Applications of Transistors*. Elsevier Science, 2016. ISBN 9781483138886. URL https://books.google.cz/books?id=g9Y_DQAAQBAJ.

[14] Resistors. URL https://learn.sparkfun.com/tutorials/resistors/all.

[15] Potentiometers. URL https://www.electronics-tutorials.ws/resistor/potentiometer.html.

[16] Condenser. URL https://www.britannica.com/technology/condenser-cooling-device.

[17] balsamiq. URL https://balsamiq.com/.

[18] Microcap12. URL http://www.spectrum-soft.com/index.shtm.

[19] The Apache Software Foundation. Apache maven project. URL https://maven.apache.org.

[20] Javafx. URL https://openjfx.io/.

[21] Inc. Fazecast. jserialcomm. URL https://fazecast.github.io/jSerialComm.

[22] Jackson project. URL https://github.com/FasterXML/jackson?fbclid=IwAR2fMfvsLSVkoFg0P7uqYWeIAgWDH5QbpqD6WpPcMNF8ZQtGuzP8JS6RLVg.

[23] Úvod do json. URL https://www.json.org/json-cz.html.

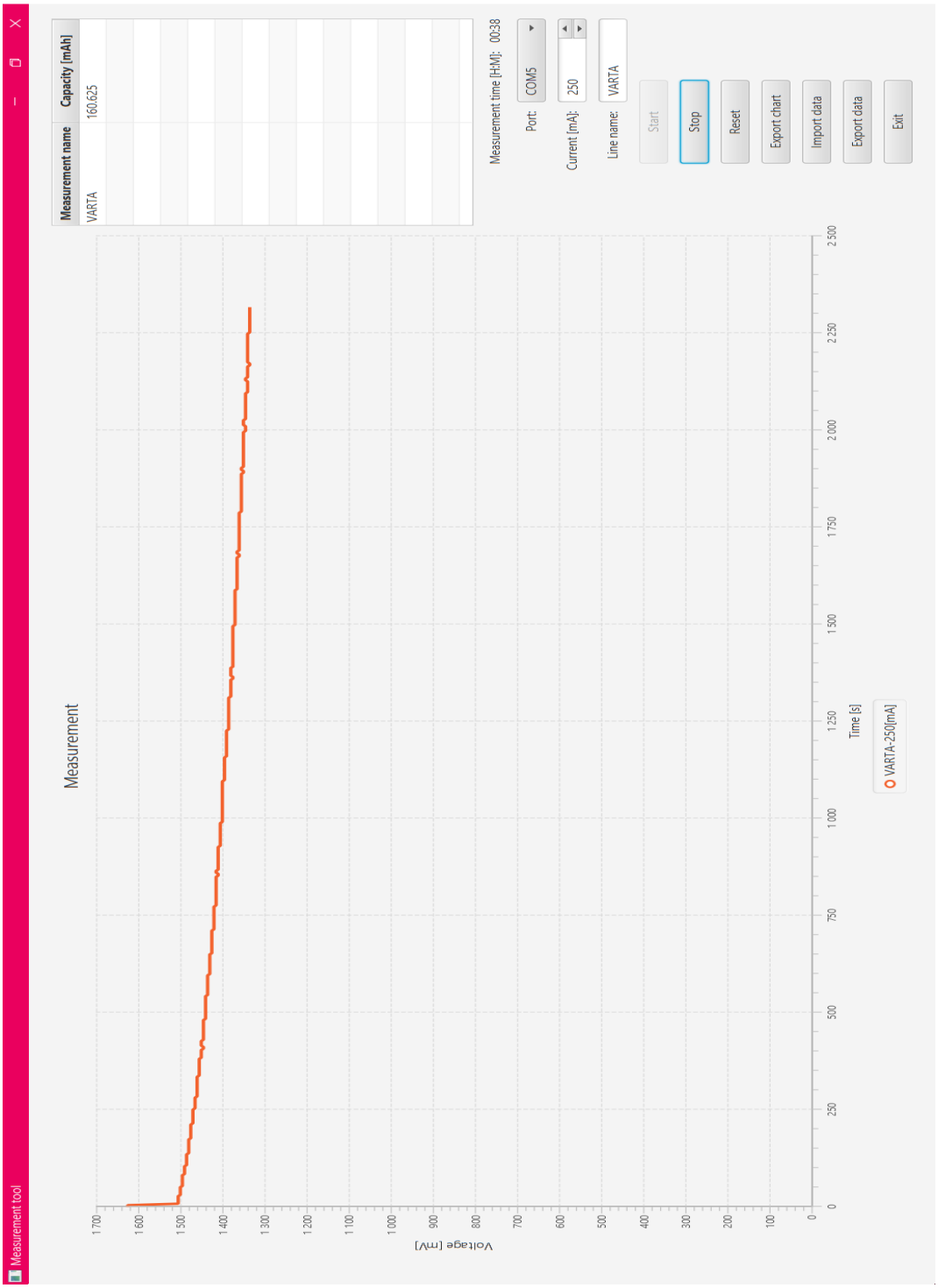
[24] Singleton class in java. URL https://www.geeksforgeeks.org/singleton-class-java/?fbclid=IwAR08utIOHM7uYJrXEZFR7LLe_hRjccGldg2K7AKAXu7U6qz-FiNQ_jFbokU.

Přílohy

Charakteristiky naměřených baterií

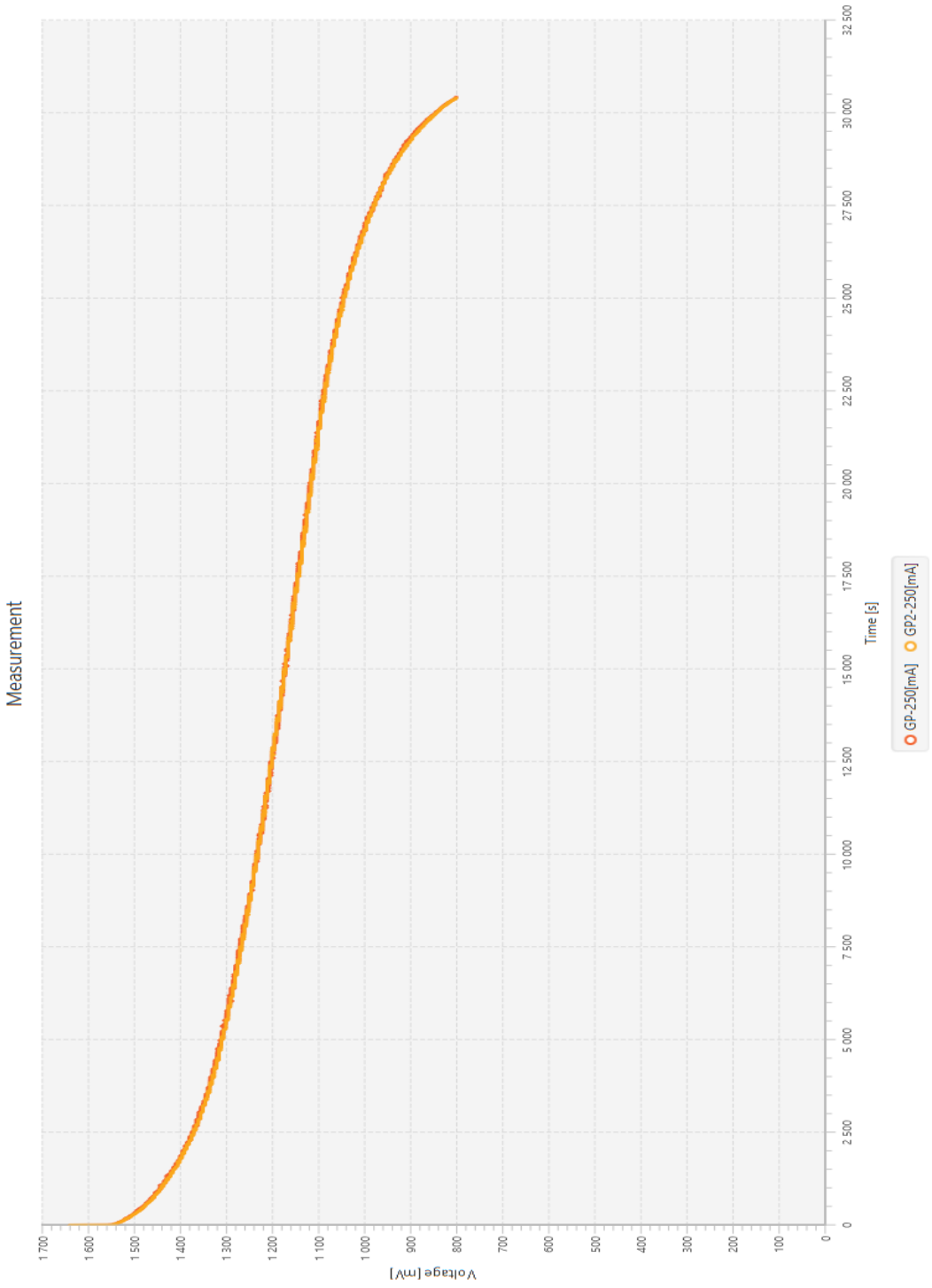
Pro překlad a spuštění aplikace je zapotřebí mít v počítači nainstalovanou Javu alespoň ve verzi 11. V adresáři projektu je umístěn Maven wrapper, maven tedy není třeba instalovat. Pro usnadnění přeložení programu je dále přiložen soubor build.cmd. Po jeho spuštění se provede překlad aplikace a výsledný .jar soubor se umístí do podadresáři target v adresáři projektu. Vytvořený soubor .jar obsahuje všechny knihovny potřebné pro běh aplikace a lze tedy aplikaci rovnou spustit. K usnadnění spuštění je přiložen soubor run.cmd. K aplikaci na PC je byla vygenerována Java dokumentace. Je přiložena v adresáři projektu ve složce JavaDoc.

Po spuštění aplikace se zobrazí hlavní okno aplikace. Aplikace kromě dialogových oken, které slouží pro zobrazení chyb, varování nebo specifikování cesty nemá žádné další okna. Hlavní okno aplikace obsahuje graf zobrazující vybíjecí charakteristiky. V pravé části okna se nachází tabulka se záznamem celkové kapacity baterií (načtených do aplikace nebo změřených) a tlačítka na ovládání programu. Měření lze spustit tlačítkem *Start* pouze po připojení Arduina se zařízením do USB a vybráním správného portu. Spuštěné měření lze kdykoli zastavit tlačítkem *Stop*. Zaznamenaná měření lze hromadně uložit tlačítem *Export Data* jako samostatné soubory s názvem <název>-<hondota vybíjecího proudu [mA]>.json, kdy název je vytvořen automaticky. Pokud již soubor s takovým názvem existuje, bude uživateli zobrazen potvrzovací dialog s otázkou, zda chce daný soubor přepsat. Tyto soubory se ukládají do složky Data v kořenovém adresáři projektu. Složka Data se automaticky vytvoří při prvním uložení zaznamenaných měření. Každé měření lze samostatně nahrát zpět do aplikace pomocí tlačítka *Import data*. Po vybrání tohoto tlačítka bude uživateli zobrazen dialog, ve kterém je třeba vybrat cestu k danému souboru měření, který bude nahrán do aplikace. Dále lze také zvolit uložení grafu pomocí tlačítka *Export chart*, kdy se uloží obrázek grafu do adresáře Charts, který se nachází, popřípadě vytvoří při prvním uložení obrázku grafu, v adresáři projektu. Obrázek grafu měření se uloží jako <název>.png, kde název obrázku si uživatel volí sám. Graf s tabulkou lze vymazat tlačítkem *Reset* a program lze ukončit tlačítkem *Exit*.

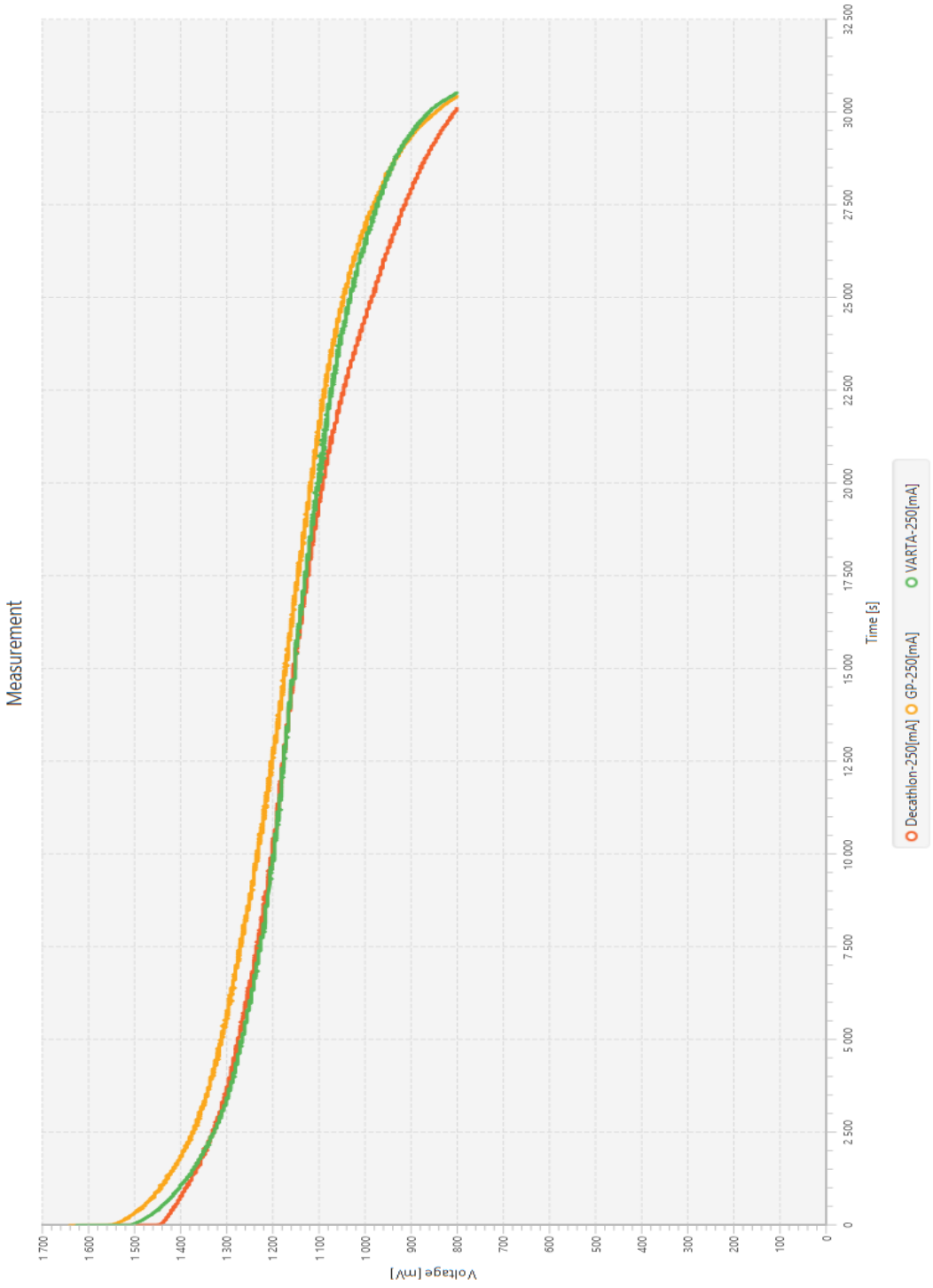


Obrázek 10.1: Uživatelské prostředí

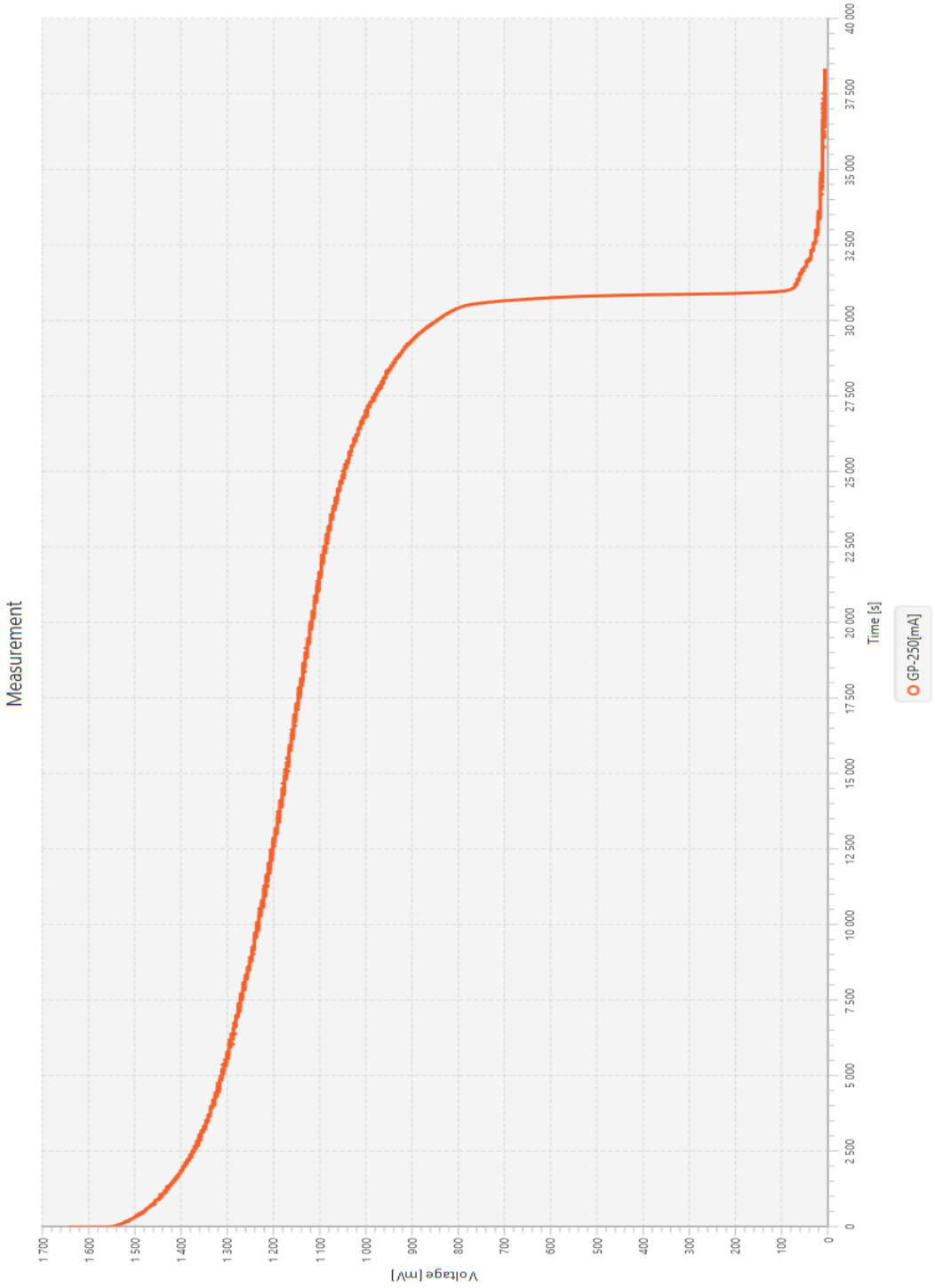
Charakteristiky naměřených baterií



Obrázek 10.2: Porovnání dvou baterií stejné značky



Obrázek 10.3: Porovnání více značek baterií



Obrázek 10.4: Charakteristika plně vybitého článku