

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra mikroelektroniky

Zařízení pro charakterizaci zdrojů
stejnoseměrného napětí

květen 2016

Bakalant:
Vedoucí práce:

Karel Zdražil
Ing. Adam Bouřa, Ph.D.

Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce a konzultanta a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry.

Datum: 16. 5. 2016

.....
podpis bakalanta

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Karel Zadražil**

Studijní program: Komunikace, multimédia a elektronika
Obor: Síťové a informační technologie

Název tématu: **Zařízení pro charakterizaci zdrojů stejnosměrného napětí**

Pokyny pro vypracování:

1. Proveďte rešerši a seznamte se s problematikou zdrojů stejnosměrného napětí (chemické články, palivové články, elektronické obvody) a způsoby monitorování jejich stavu a základních elektrických parametrů.
2. Na základě získaných poznatků zvolte vhodné technické řešení, navrhnete a realizujete zařízení pro monitorování okamžitého stavu připojeného zdroje napětí (vnitřní napětí, vnitřní odpor, maximální možný výkon do přizpůsobené zátěže) a pro monitorování stavu na připojené zátěži (napětí, proud, okamžitý výkon, celková spotřebovaná energie za daný čas).
3. Sestavte testovací pracoviště, proveďte měření a kriticky zhodnoťte dosažené parametry vašeho řešení. Na základě výsledků testování případně navrhnete další možná vylepšení.

Seznam odborné literatury:

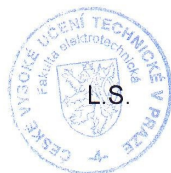
- [1] Krejčířík, A.: *Napájecí zdroje I: Základní zapojení analogových a spínaných napájecích zdrojů*. 1. vyd. Praha: BEN, 1996, 341 s. ISBN: 80-860-5602-3.
- [2] Billings, K.; Morey, T.: *Switchmode powersupply handbook*. McGraw-Hill, 2011. ISBN: 978-0-07-163971.
- [3] Lenk, R.: *Practical design of power supplies*. IEEE press 2005. ISBN: 978-0-471-75045-1.
- [4] Vedral, J.; Fischer, J.: *Elektronické obvody pro měřicí techniku*, Vydavatelství ČVUT, Praha, 2004. ISBN: 80-01-02966-2.

Vedoucí: Ing. Adam Bouřa, Ph.D.

Platnost zadání: do konce letního semestru 2016/2017



prof. Ing. Boris Šimák, CSc.
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 21. 12. 2015

Seznam použitých zkratk:

CMOS	technologie výroby polovodičových čipů (Complementary Metal–Oxide–Semiconductor)
DPS	deska plošných spojů
EEPROM	elektronicky smazatelná paměť (Electrically Erasable Programmable Read-Only Memory)
LCD	displej z tekutých krystalů (Liquid Crystal Display)
MCU	řídící jednotka (Machine Control Unit)
OZ	operační zesilovač
PC	Personal Computer
PWM	pulzně šířková modulace (Pulse Width Modulation)
SRAM	statická rychlá paměť - nepotřebuje periodickou obnovu (Static Random Access Memory)
THT	osazování plošných spojů součástkami s drátovými vývody (Through-Hole Technology)
UART	univerzální asynchronní sériové rozhraní UART (Universal Asynchronous Receiver and Transmitter)
USB	Universal Serial Bus

Anotace:

Tato bakalářská práce pojednává o stejnosměrných zdrojích elektrické energie a o vývoji a výrobě měřicího přístroje pro tyto zdroje a jejich zátěže. Přístroj měří napětí, proud a vnitřní odpor. Řízení obstarává Arduino. Úpravu měřeného napětí zajišťuje soustava operačních zesilovačů. Naměřené hodnoty, jako napětí, proud, vyrobená/spotřebovaná energie, jsou zobrazovány na displeji přístroje.

Klíčová slova:

Arduino, měření proudu, měření napětí, měření spotřeby, stejnosměrné zdroje

Summary:

This bachelor thesis deals about DC power sources and about development and fabrication of measuring instruments for these resources. This device measures current, voltage and internal resistance. The device is controlled by Arduino. Operational amplifiers provide adjustment of the measured voltage. Measured values like voltage, current, produced/consumed energy are displayed on LCD screen.

Index Terms:

Arduino, current measurement, voltage measurement, the power consumption measurement, DC power sources

Seznam obrázků

Obrázek 1: Návrh DPS vytvořený v programu Eagle.....	13
Obrázek 2: Pohled na pracoviště při testování na vodíkovém článku.....	26
Obrázek 3: Zobrazení napětí a proudu na displeji.....	28
Obrázek 4: Použité NiMH články.....	29
Obrázek 5: Testování na Li-ion baterii a reálném spotřebiči – LED svítelně.....	30
Obrázek 6: Testování na solárním článku.....	31

Seznam grafů

Graf 1: Průběh proudu a napětí při generování vodíku.....	27
Graf 2: Graf pro generování elektrické energie v článku.....	27
Graf 3: Klouzavý průměr grafu generovaného proudu a napětí.....	28
Graf 4: Zachycená charakteristika NiMH článku.....	29
Graf 5: Experiment s vybíjením Li-ion.....	30
Graf 6: Výstup testování na solárním článku.....	31

Seznam schémat

Schéma 1: Vnitřní odpor zdroje.....	3
Schéma 2: Zapojení měřicího odporu a AD620.....	6
Schéma 3: Blokové schéma.....	8
Schéma 4: Ochrana na vstupu.....	9
Schéma 5: Zapojení integrovaného stabilizátoru.....	9
Schéma 6: Zdroj záporného napětí pro operační zesilovače.....	9
Schéma 7: Zapojení diod MCU.....	10
Schéma 8: Vstupní dělič a sledovač napětí.....	11
Schéma 9: Zapojení nevyužitého OZ.....	11

Obsah

1 Úvod.....	1
2 Cíle práce.....	2
3 Teoretický rozbor.....	3
3.1 Měření zdrojů napětí.....	3
3.2 Galvanické články primární.....	4
3.2.1 Zinko-uhlíkové články.....	4
3.2.2 Zinko-chloridové články.....	4
3.2.3 Alkalické články.....	4
3.3 Galvanické články sekundární.....	5
3.3.1 Palivové články.....	5
3.3.2 Niklo-kadmiové články.....	5
3.3.3 Niklo-metalhydridové akumulátory.....	5
3.3.4 Li-ion.....	5
3.3.5 Olověné akumulátory.....	5
3.4 Zhodnocení.....	6
3.5 Použité prostředky.....	7
3.5.1 Arduino.....	7
3.5.2 Displej.....	7
4 Návrh a realizace.....	8
4.1 Aplikace.....	8
4.2 Napájení.....	8
4.2.1 Možnosti napájení.....	8
4.2.2 Konečný návrh napájení.....	9
4.3 Návrh měření proudu a napětí.....	10
4.4 Měření napětí.....	11
4.4.1 Vstupní dělič.....	11
4.4.2 Napěťový sledovač.....	11
4.5 Měření proudu.....	12
4.6 Přepínání rozsahů.....	12
4.7 DPS.....	12
4.7.1 Zdířky.....	13
4.7.2 Napájecí jack konektor.....	13
4.7.3 Osazení.....	13
4.8 Program.....	14
4.8.1 Obsluha přístroje.....	14
4.8.2 Hlavička a knihovny.....	14
4.8.3 Deklarace globálních proměnných.....	15
4.8.4 Inicializace a funkce „setup“.....	17
4.8.5 Funkce „loop“.....	18
4.8.6 Funkce pro změnu rozsahu.....	21
4.8.7 Funkce pro přepisování displeje.....	23
4.8.8 Funkce pro výpis do sériové linky.....	24
4.8.9 Kompilace a flashování.....	25
5 Testování.....	26
5.1 Měření s vodíkovým článkem.....	26
5.2 NiMH články.....	29

5.3 Zatěžovací rezistory.....	29
5.4 Měření na Li-ion článku.....	30
5.5 Solární článek.....	31
6 Možnosti dalšího vylepšení.....	32
6.1.1 Detekce polaritý napětí a směru proudu.....	32
6.1.2 Elektroměr.....	32
6.1.3 Konektivita a záznam.....	32
6.1.4 Jas.....	32
6.1.5 Kontrast.....	33
7 Závěr.....	34
7.1 Technické údaje:.....	34
7.2 Poznámky k použití.....	34
Seznam použité literatury.....	36
Seznam příloh:.....	36
Seznam příloh na CD:.....	36

1 Úvod

Inspirací pro vývoj tohoto přístroje byl měřič určený pro použití s vodíkovým palivovým článkem. Ten je omezen pouze na použití spolu s PC a dnes již nepodporovaným operačním systémem Windows XP. Vyžaduje instalaci softwaru dodaného výrobcem a nemůže pracovat autonomně.

Proto jsem se rozhodl navrhnout měřicí přístroj, který nebude trpět výše zmíněnými nedostatky a bude také použitelný ve výuce. V případě dalšího zpracování naměřených hodnot není závislý ani na použitém operačním systému.

Tomuto návrhu předcházela můj individuální projekt a toto je již druhá, zdokonalená verze tohoto přístroje. Dále se o něm ještě zmíním.

Práce se ve své teoretické části zabývá zdroji stejnosměrného napětí a problematikou měření. V praktické části je podrobně popsán návrh i konstrukce vlastního zařízení.

Svůj postup jsem se snažil zdokumentovat tak, aby čtenář byl schopný mé zařízení modifikovat a postavit si vlastní. Zvláště důkladně jsem komentoval software, abych tak usnadnil jeho případnou úpravu. Předposlední kapitola se věnuje testování měřicího zařízení s reálnými zdroji a různými zátěžemi.

2 Cíle práce

Cílem je nastudovat charakter běžně používaných zdrojů stejnosměrného napětí a sestavit zařízení pro jejich charakterizaci, včetně jeho návrhu a vytvoření firmware, který bude toto zařízení řídit. Ze zadání práce kromě požadavku na měření napětí, proudu, výkonu a elektrické energie také vyplývá nutnost měřit vnitřní odpor.

Dále jsme si spolu s vedoucím práce položili za cíl integrovat automatické přepínání rozsahů pro zvýšení přesnosti pro malá napětí. Dále bylo vhodné využít součástky a ověřená zapojení z individuálního projektu, především přístrojový zesilovač a nábojovou pumpu. Obsluhu také velmi zjednodušilo usměrnění signálu před jeho měřením, proto jsem se rozhodl i do tohoto přípravku tento prvek použít.

Další požadavky vyplývají z principu měření jako takového. Zařízení nesmí výrazným způsobem zatěžovat měřený obvod a tím znemožnit jeho správnou funkci a zkreslit měření.

Závěrem je také nutné zařízení otestovat na reálných zdrojích napětí, aby mohla být potvrzena funkčnost a ověřeno, že měření nenaruší nějaký opomenutý jev.

3 Teoretický rozbor

3.1 Měření zdrojů napětí

Všechny reálné zdroje stejnosměrného napětí si lze představit jako ideální zdroj napětí se sériově zařazeným rezistorem. Napětí tohoto ideálního zdroje se nazývá elektromotorické a u nezátženého zdroje je rovno svorkovému. Situace je patrná ze schématu níže. Pokud chceme toto napětí (U_0) změřit, je vhodné použít měřicí přístroj s maximálním vstupním odporem. Při připojení měřicího přístroje vznikne odporový dělič napětí, který je tvořen vnitřním odporem zdroje napětí (R_i) a vnitřním odporem přiloženého voltmetru (R_z).

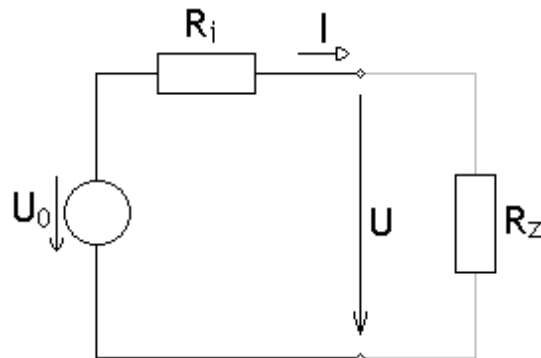


Schéma 1: Vnitřní odpor zdroje [1]

Napětí které měříme je potom dané vztahem 1:

$$U = U_0 \frac{R_z}{R_i + R_z} \quad (1)$$

Z toho je patrné, že s klesajícím odporem voltmetru R_z se začne negativně projevovat vnitřní odpor R_i a měřené napětí U bude klesat. Proto je potřeba, aby byl vstupní odpor voltmetru řádově větší, než vnitřní odpor měřeného zdroje, čímž se minimalizuje vzniklá chyba.

Obdobně je tomu v případě měření proudu, kdy se proud procházející uzavřenou smyčkou opět řídí ohmovými zákony. Platí zde nepřímá úměra - čím menší odpor do obvodu vložíme, o to méně snížíme proud, který budeme následně měřit. S tím je potřeba při návrhu ampérmetru počítat a volit odpor řádově menší, než je odpor spotřebiče. Jedině pak je možno jej při měření zanedbat.

Měření vnitřního odporu lze provést například dvojitým měřením napětí a proudů procházejících zdrojem. Výsledný odpor se pak vypočítá jako rozdíl napětí podělený rozdílem proudů. Změny procházejícího proudu lze docílit například připnutím zatěžovacího rezistoru. Tak je tomu i v tomto případě. Výsledný vnitřní odpor zdroje se potom vypočítá z rozdílů napětí a proudů s připnutým a nepřipnutým rezistorem:

$$R = \frac{U_1 - U_2}{I_2 - I_1} \quad (2)$$

Je zřejmé, že tento zatěžovací odpor je opět nutné volit s ohledem na měřené zdroje. Příliš

vysoký odpor v korespondenci s vnitřním odporem zdroje nevyvolá měřitelný úbytek napětí a naopak. S příliš malým bude svorkové napětí blízké nule, což způsobí dva problémy. Prvním je změření nízkého napětí s dostatečnou přesností pro určení proudu. V případě připojení zdroje s nízkou hodnotou odporu potom hrozí zničení zdroje i měřicího přístroje nadměrným proudem.

3.2 Galvanické články primární

Primární články slouží především pro jednorázové použití. Ačkoliv se dnes dají sehnat speciální nabíječky alkalických baterií, jejich použitelnost je velmi omezená. Pro opakované použití je totiž velmi důležité, jakým proudem byly během používání vybíjeny a jak dlouho následně skladovány. Například baterie z dálkového ovladače proto nelze již znovu použít, ale baterie vybíjené vysokým proudem a bezprostředně poté znovu nabité mohou znovu nabídnout přes 50 % své původní kapacity. [2] Nejčastěji se dnes používají právě alkalické a pak zinko-chloridové články.

3.2.1 Zinko-uhlíkové články

V minulosti se hojně používaly zinko-uhlíkové články, od kterých se dnes postupně upouští. Děje se tak z důvodu vznikající volné vody při jejich vybíjení a nebezpečí následného vytečení článku. Pokud je takto postižený článek umístěn ve spotřebiči, znamená to obvykle destrukci jeho plošného spoje. Jejich jmenovité napětí je 1,5 V. Kapacita tužkového článku se pohybuje kolem 800 mAh (pro konečné napětí 0,9 V a proud 0,1 A) a oproti alkalickým ale i zinko-chloridovým článkům, není schopen dodávat tak vysoký proud. [3]

3.2.2 Zinko-chloridové články

Zinko-chloridové články dosahují napětí 1,62 V naprázdno u čerstvého článku. Jmenovité napětí je stejné – 1,5 V. Při zatížení 10Ω rezistorem klesne napětí na 1,3 V, což odpovídá vnitřnímu odporu řádově jednotek ohmů. Na rozdíl od zinko-uhlíkových článků, který je konstruován jako suchý a během vybíjení se v něm vytváří voda, je zinko-chloridový článek vlhký a voda se v něm během používání spotřebovává. Vybitý článek je proto suchý, s čím souvisí i postupné zvyšování vnitřního odporu. Kapacita se pohybuje také okolo 800 mAh tužkového článku.

3.2.3 Alkalické články

Mají stejné jmenovité i maximální napětí jako předchozí dva jmenované zdroje, avšak díky odlišné konstrukci je využitelnost aktivní hmoty článku mnohem vyšší. Jejich kapacita je zhruba 1,6 Ah a vnitřní odpor řádově menší, tedy desetiny ohmu. Tomu zároveň odpovídá i odlišná zatěžovací charakteristika, která je díky tomuto daleko plošší.

Lithiové primární články mají ještě nižší vnitřní odpor, než alkalické baterie stejné kapacity. Jejich napětí naprázdno se pohybuje kolem 1,7 nebo 3,6 V.

3.3 Galvanické články sekundární

3.3.1 Palivové články

Spadají do kategorie galvanických článků, protože mění chemickou energii na elektrickou. Slučují palivo s kyslíkem. Palivem může být například vodík, methanol či ethanol, okysličovadlem čistý kyslík. Tyto články mohou pracovat nepřetržitě, je však nutno zajistit, aby odpadní produkty nezůstávaly v článku a nezmenšovaly tak aktivní plochu. Nejčastější konstrukční řešení se skládá z polopropustné membrány, nebo elektrolytu obklopeného dvěma elektrodami. Ze strany kde dodáváme palivo je záporně nabitá anoda a na straně paliva katoda s kladným nábojem.

Napětí palivového článku má teoretické maximum 1,23 V. Reálné články běžně dosahují napětí 0,5 – 1 V a proudu 0,5 – 2 A/cm² aktivní plochy. Běžné komerčně nabízené články mají výkon kolem 0,5 W/cm². Vnitřní odpor laboratorního 0,3W článku je řádově desítky až stovky ohmů, v závislosti na provozním režimu (generace vodíku/výroba elektřiny) a na míře nasycení aktivní plochy vodou.ku/vý

3.3.2 Niklo-kadmiové články

Jejich jmenovité napětí je 1,2 V, vnitřní odpor se potom pohybuje řádově v setinách ohmu a kapacita tužkové baterie běžně dosahuje 1 Ah.

3.3.3 Niklo-metalhydridové akumulátory

Mají stejné jmenovité napětí, ale zhruba dvakrát vyšší kapacitu, než niklo-kadmiové články stejné velikosti, přičemž vnitřní odpor bývá více než dvojnásobný. Vše záleží na výrobním postupu, avšak pro pohon zařízení jako jsou akumulátorové vrtačky nejsou tyto články příliš vhodné.

3.3.4 Li-ion

Dalším typem dnes hojně používaných akumulátorů jsou lithiové – ty našly své uplatnění v drobné spotřební elektronice, jako jsou mobilní telefony, mp3 přehrávače a podobně. Jejich jmenovité napětí je 3,6 V. Jsou velmi citlivé na přebíjení a naopak hluboké vybíjení. Nabíjení nad 4,4 V je velmi poškozující a stejně tak vybití pod 2,5V je nenávratně ničí. [4] Také je velmi důležité dodržet maximální nabíjecí a vybíjecí proudy vzhledem k vyššímu vnitřnímu odporu, než v případě NiCd akumulátorů. Zmíněné nevýhody jsou vykoupené vysokou kapacitou na hmotnost ale i jednotku objemu. Baterie o velikosti AA bude mít kapacitu okolo 3500 mAh. V některých spotřebičích byly nahrazeny lithium-polymerovými bateriemi, které mají menší vnitřní odpor a tak snesou větší proudové zatížení.

3.3.5 Olověné akumulátory

Poslední druh akumulátorů, který zmíním je olověný. Mají extrémně nízký vnitřní odpor, řádově setiny ohmu, což je přímo předurčuje pro různé vysoce energeticky náročné aplikace, jako jsou třeba zálohovací UPS jednotky. Nominální napětí článku je 2,1 V a kapacita akumulátoru vážící 1kg bude cca 4 Ah.

3.4 Zhodnocení

Často se nesetkáme jen se samotnými jednotlivými články. Pokud je potřeba dosáhnout vyššího napětí, spojují se sériově. Olověné akumulátory nejčastěji po třech až dvanácti kusech do baterií se jmenovitým napětím 6 V až 24 V. Lithiové po dvou nebo třech kusech s nominálním napětím baterie 7,2 V nebo 10,8 V. Vzhledem k tomu, že se jedná o laboratorní měřicí přístroj, bylo maximální napětí s ohledem a dostatečnou rezervu stanoveno na 25 V. Také bylo nutné zvolit nejvyšší proudový rozsah, vzhledem k měřeným zdrojům, jejich zatížitelnosti a vnitřním odporům je hodnota 1,3 A pro většinu z nich dostatečná. Měřicí odpor R1 (viz schéma 2) má své maximum 6 A.

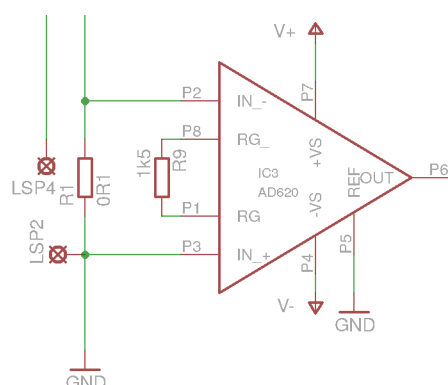


Schéma 2: Zapojení měřicího odporu a AD620

Tato hodnota vyplývá ze vztahu 3:

$$I_{max} = \sqrt{\frac{P_{max}}{R}} \quad (3)$$

Pro případné nutnosti navýšení proudového rozsahu až do této hodnoty stačí vyměnit rezistor R9 za větší, přepočítat příslušné konstanty a nahrát nový program do Arduina. Obdobný postup lze aplikovat i pro zvýšení napěťového rozsahu, kdy maximální napětí je dáno maximálním bezpečným stejnosměrným napětím, tedy 100V.

Kvůli velkému rozptylu vnitřních odporů jsem přistoupil k použití vícero rezistorů. Pro měření touto metodou by bylo ideální, kdyby velikost zatěžovacího rezistoru byla stejná nebo alespoň srovnatelná s velikostí vnitřního odporu zdroje. Proto jsem se rozhodl pro hodnoty 15 Ω , 43 Ω a 100 Ω . Nižší hodnoty odporů si vzhledem k přípustnému napětí nemůžeme dovolit. Tekl by příliš velký proud, kdy hrozí zničení relé, případně připojeného rezistoru.

3.5 Použité prostředky

3.5.1 Arduino

Díky použití Arduina k odečtu hodnot se toto zařízení stává velmi flexibilním. Způsob, jakým se budou naměřená data zpracovávat, lze velmi jednoduše upravit změnou kódu a nahráním nového programu. Data, která se budou zobrazovat na displej a vypisovat do sériového rozhraní, se dají stejně jednoduše vybrat a uzpůsobit.

Při volbě Arduina byly rozhodujícími parametry především počet I/O pinů (včetně těch analogových) a fyzická velikost. Jelikož bylo nutné na desku umístit poměrně velký displej, tak by rozměrné Arduino znamenalo výrazné zvětšení desky, což by se mimo jiné nepříznivě projevilo na ceně při zakázkové výrobě této desky. Z toho důvodu jsem vybral Arduino Pro Mini, které má v základu vyvedeno 16 vstupně/výstupních pinů a je relativně malé. Některé klony mají navíc možnost vyvedení dalších pinů. Toto Arduino je osazené Atmegou328p, která má 32 kilobytů flash paměti pro program, 2 kilobyty SRAM pro proměnné a 1024 bytů EEPROM paměti, kterou využívám pro uložení nastavení. Taktovací frekvence 16 MHz je pro toto nasazení více než dostačující. [5]

Jak již bylo zmíněno, Arduino samotné je napájeno stabilizovaným napětím 5 V. Jsou k němu připojena dvě tlačítka. Pull-up rezistory nejsou potřeba – ty jsou již obsažené v samotném MCU a programově zapnuty. Stejně tak ošetření zákmitů tlačítek je řešeno softwarově. Samo Arduino přímo ovládá přepínání rozsahů a spínání cívky jazýčkového relé.

3.5.2 Displej

Samozřejmě je také přímo k Arduinu připojen LCD displej s řadičem, který je kompatibilní s čipem HD44780. Takovéto displeje se ovládají osmibitovými instrukcemi, ale podporují i čtyřbitový režim, kdy se všechna data i instrukce odesílají ve dvou krocích. V takovém případě se potom používá pouze horních čtyř bitů. Ty nepoužité je možné uzemnit jako prevenci před pronikáním rušení. Tento režim je výhodnější kvůli úspoře I/O pinů. Kromě toho má tento řadič ještě pin RS, kterým se přepíná mezi vkládáním instrukcí, nebo dat a E sloužící pro povolení přenosu. Pin, který nese označení R/W nebyl použit. Funguje jako přepínač mezi zápisem na displej a zpětným čtením z něj. Stejně jako nevyužité datové piny je i on uzemněn. [6]

Za povšimnutí stojí vstup označený V0, kterým se analogově (napětím 0 – 5 V) nastavuje kontrast. Ten je regulován taktéž softwarově PWM modulací. Jelikož výrobce počítá s analogovou hodnotou napětí, je výsledný analogový výstup získán RC článkem složeným z odporu R8 a kondenzátoru C6.

Displej je podsvětlen. Jelikož jsem katalogové listy k tomuto konkrétnímu displeji nenašel a některé podobné displeje počítají s napětím pro podsvětlení 4,2 V, rozhodl jsem se před displej zařadit rezistor. Jeho hodnotu jsem zjišťoval empiricky. Při pohledu na intenzitu podsvícení se zdá, že tento konkrétní model je koncipovaný na provoz pro 5 V. Přesto jsem tam rezistor nechal. Ve dne není podsvícení potřebné vůbec a v noci pak nepůsobí zbytečně rušivě.

4 Návrh a realizace

Návrh zapojení vychází z ověřeného zapojení, které jsem realizoval v individuálním projektu. Oproti němu doznalo některých změn, jmenovitě zcela odlišně řešené napájení, automatická změna rozsahů, dvě tlačítka pro ovládání, možnost měření vnitřního odporu a napěťový sledovač na vstupu. Tyto změny byly potřeba kvůli nutnosti vyhovět zadání této bakalářské práce, nebo v návaznosti na takovéto úpravy. Celkové schéma naleznete v příloze I, blokové schéma pod tímto textem.

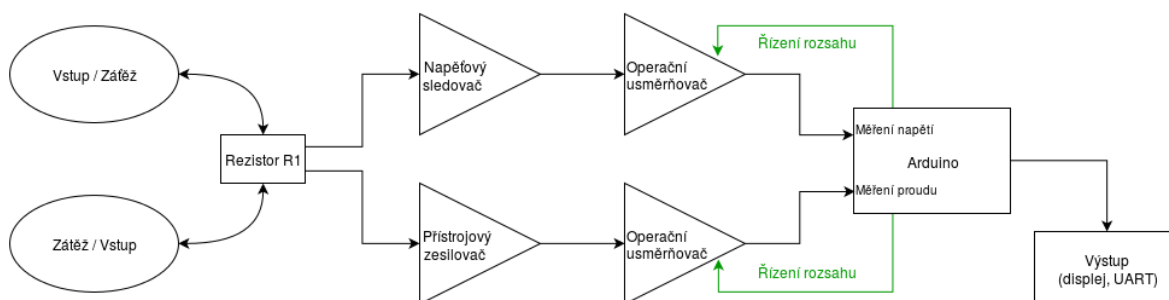


Schéma 3: Blokové schéma

4.1 Aplikace

Při návrhu byl kladen důraz především na snadnost obsluhy. Proto byla na vstupu použita dvojice integrovaných usměrňovačů, díky nimž lze zdroj i spotřebič zapojit s libovolnou polaritou, stejně tak zaměnit zdroj se spotřebičem.

Aby nebylo potřeba napájet OZ z externího zdroje, je pro získání záporné větve použita nábojová pumpa.

Displej kromě přeneseného náboje zobrazuje zároveň i údaje o aktuálním napětí a proudu. Jak již bylo zmíněno, přeprogramováním Arduina může zobrazovat též například aktuální příkon zátěže. Přístrojový zesilovač AD620 zajišťuje snímání proudu a předzesílení. OZ LM358N je zapojen jako napěťový sledovač pro zvýšení vstupního odporu přístroje na měřicích svorkách.

Měřič je konstruován na proud až 1,3 A a napětí až 25 V. Kromě využití s vodíkovými či ethanolovými palivovými články se tak též hodí pro měření i s velkou spoustou jiných stejnosměrných zdrojů.

4.2 Napájení

4.2.1 Možnosti napájení

Jelikož je na vstupu použit odporový dělič napětí, není nutné, aby napájecí napětí OZ převyšovalo vstupní. Většina komponent je proto napájena jednotným napětím 5V, což zároveň díky současné nízké spotřebě celého zařízení dává možnost napájení přímo z portu USB. Napětí z USB je možno vyvést například prostřednictvím USB – UART převodníku,

který zároveň zajistí zachytávání naměřených dat. Nevýhodou takového řešení je však galvanické propojení PC s měřeným napětím. Konkrétně se jedná o svorku INP.2-B spojenou s pájecí ploškou LSP2 na které je sdílena zem (viz schéma 2). U síťově napájených aplikací je v takovém případě naprosto nezbytné ujistit se, že nemůže jakkoli kolidovat připojené napětí se zemí počítače. Lepším řešením je potom použití USB rozbočovače s galvanicky oddělenými porty či USB – UART převodníku bez galvanické vazby obou výstupů. Pokud budou k měření použity pouze galvanické články bez dodatečného zemnění, nebezpečí proniknutí tohoto napětí do USB počítače nehrozí i při případném proražení vstupního sledovače napětí. Přístrojový zesilovač sledující proud oproti tomu má oba své vstupy téměř na nulovém potenciálu. Je tomu tak díky výkonovému rezistoru R1.

4.2.2 Konečný návrh napájení

Protože je deska koncipována spíše pro autonomní provoz, je osazena klasickým 2,1mm jack konektorem pro externí zdroj napětí 8 až 12 V. Sériově s ním je zařazena ochranná dioda D1, která je při normálním provozu orientovaná v propustném směru. Slouží jako ochrana proti nechtěnému přepólování. Za touto diodou se nachází kondenzátor C5, který má za úkol případné vyhlazení napájecího napětí.

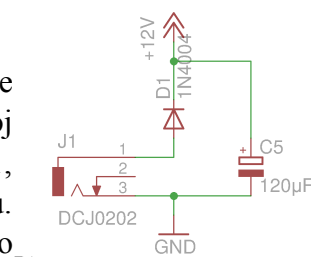


Schéma 4: Ochrana na vstupu

Plné napájecí napětí je poté přivedeno do pevného lineárního regulátoru napětí 7805 v doporučeném zapojení. To zahrnuje blokovací kondenzátory C3 a C4. Kvůli větší stabilitě napájecího napětí je na 5V větev přidán kondenzátor C7. Aby se nestalo, že se tento poměrně velký kondenzátor vybijí skrz integrovaný stabilizátor napětí, je doplněna další ochranná dioda D6, která tento stabilizátor ochrání v případě odpojení napájení. Také se uplatní při napájení přímo prostřednictvím Arduina, jelikož by i v tomto případě vznikl opačný napěťový potenciál na stabilizátoru.

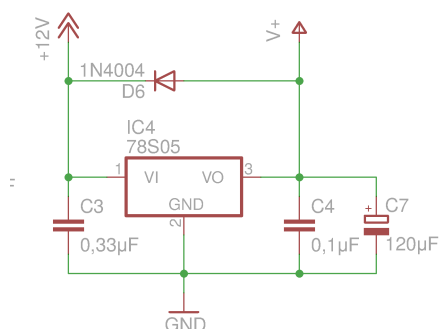


Schéma 5: Zapojení integrovaného stabilizátoru

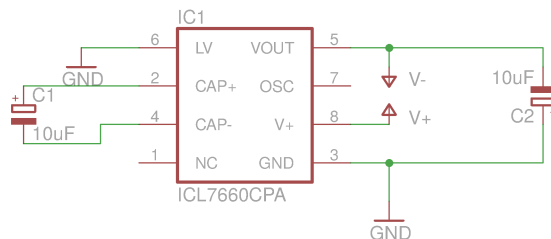


Schéma 6: Zdroj záporného napětí pro operační zesilovače

Napájecí řetězec uzavírá ICL7660. Jedná se o nábojovou pumpu vyrábějící shodné napětí, ale opačné polarity pro OZ, které ke správné činnosti potřebují symetrické napájecí napětí.

Za tímto účelem používá kondenzátor C1. Kondenzátor C2 opět slouží ke stabilizaci takto získaného napětí. Vzhledem k nízké spotřebě byla zvolena menší hodnota, než v případě již zmíněných C5 a C7.

Všechny komponenty, které si vystačí s jednoduchým napájením, tedy displej a Arduino jsou připojeny na 5 V ze stabilizátoru. Arduino má svůj vlastní 5V stabilizátor, ten by však pro napájení celé sestavy nebyl dostatečně výkonný, a proto ho nevyužívám.

4.3 Návrh měření proudu a napětí

Proud je snímán z vloženého rezistoru R1, který je na jedné straně uzemněn. Díky velmi malému úbytku může být napětí snímáno jednoduše přímo ze zdírek opačné polarity. Při maximálním proudu 1 A se na odporu 0,10 Ω vytvoří napětí 0,10 V. Změna směru proudu tak nemá na přesnost měření napětí zásadní vliv. Také pro měřený obvod není vložený rezistor příliš velkou zátěží díky nízkému úbytku.

Zesílení je nastavené rezistorem R1 = 1,5 kΩ a vypočítá se podle vzorce udaného výrobcem [7]:

$$A = 1 + \frac{49400}{R} \quad (4)$$

Takto zesílené napětí je přivedeno na operační usměrňovač, který změní případnou zápornou polaritu napětí na kladnou. Usměrněné napětí je potom skrz ochranný rezistor R20 přiveden na vstup Arduino A0. Tento rezistor se při běžné činnosti neprojevívá, jelikož Arduino má na svých vstupech odpor řádově jednotky až desítky MΩ. Svou funkci splní až v případě, že by vstupující napětí mělo překročit maximální rozsah napětí, které je ještě schopný vstup akceptovat - v tomto použití -0,5 až +5,5 V. Pokud napětí vybočí z tohoto rozmezí, ochranné diody uvnitř mikrokontroléru ho upraví na bezpečnou mez a rezistor na vstupu omezí proud protékající těmito diodami. O překročení horní hranice nemusíme mít žádnou starost. OZ jsou napájeny právě pěti volty. Teoreticky by tedy mohlo napětí jít pouze do záporných hodnot. Prakticky je to také vyloučené a takováto situace může nastat maximálně při přechodových dějích. Zapojení ochranných diod je patrné z následujícího obrázku:

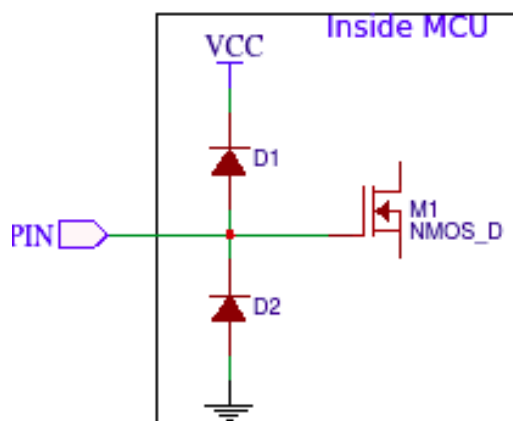


Schéma 7: Zapojení diod MCU [8]

Napětí měřeného zdroje je přímo vedeno na operační usměrňovač, odkud pokračuje prostřednictvím dalšího ochranného rezistoru na analogový převodník Arduina A1. Tento operační usměrňovač má také nastavitelný zisk.

4.4 Měření napětí

4.4.1 Vstupní dělič

Pro měření napětí je použit odporový dělič složený z rezistorů 4,7 M Ω a 1 M Ω , který zmenší vstupní napětí 5,7 krát. Bez tohoto děliče by muselo být napájecí napětí OZ ještě o 2 volty vyšší, než nejvyšší měřené, což by byl problém zejména pro volbu nábojové pumpy. Při zajištění nejvyššího rozsahu alespoň 12V bychom museli použít napájecí napětí ± 15 V. Zde použitý obvod pracuje do maximálně 10 V. Použití jiného integrovaného DC–DC měniče na vyšší napětí by bylo nepřiměřeně nákladné a vytvoření vlastního zbytečně komplikované, proto jsem se rozhodl pro tuto variantu.

Jelikož je vstupní odpor navrženého operačního usměrňovače menší než 7 k Ω , musel by mít napět'ový dělič nízkou hodnotu odporu. I za předpokladu využití faktu, že vstupní odpor tohoto operačního zesilovače je konstantní (díky tomu, že rezistory R17 a R19 jsou zapojené proti virtuální nule) by byl vstupní odpor řádově jednotky až desítky kilo ohmů. Díky použití napět'ového sledovače realizovaného pomocí OZ si můžeme dovolit odpor o dva řády vyšší.

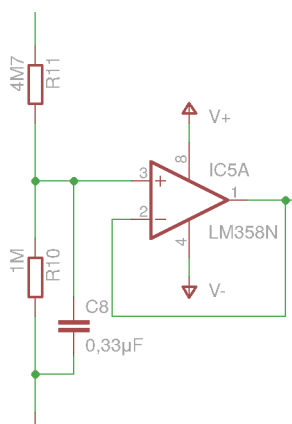


Schéma 8: Vstupní dělič a sledovač napětí

Přímo na výstup děliče je připojen kondenzátor C8. Ten by měl sloužit jako ochrana před rozkmitáním, jelikož tato zapojení bývají velmi nestabilní. Vzhledem k tomu, že v tomto případě neměl OZ tendenci se rozkmitávat, soudím že rezistor proti zemi s hodnotou 1 M Ω sám o sobě stačí a kondenzátor byl vynechán. Způsoboval by totiž velké zpoždění během ustalování nové hodnoty a to by vadilo především při měření vnitřního odporu.

4.4.2 Napět'ový sledovač

Odtud už napět'ový sledovač bere referenci pro své výstupní napětí. Jelikož jsou dva v jednom pouzdře, byly vstupy toho druhého připojeny na napájecí napětí, aby se nemohl rozkmitat a tím způsobovat rušení či přetěžovat napájecí zdroje. Výstup je potom připojen na vstup operačního usměrňovače. Jelikož je výstup velmi tvrdý (výstupní odpor se blíží nule) není třeba brát v úvahu vstupní odpor operačního usměrňovače tvořeného dvěma OZ IC2A a IC2B.

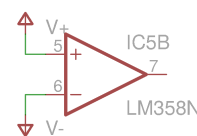


Schéma 9: Zapojení nevyužitého OZ

Ty jsou osazené kvůli případnému usměrnění záporného napětí, jelikož Atmega328p osazená v Arduinu by záporné napětí nejenže nezměřila, ale ani nesnesla. Napětí, které odchází do Arduina na deseti bitový převodník je ještě pojištěné rezistorem R21 pro případ, kdyby se z jakéhokoliv důvodu na výstupu usměrňovače objevilo jiné, než povolené napětí.

4.5 Měření proudu

Pro přímé snímání napětí na rezistoru je použit přístrojový zesilovač AD620. Jeho hlavní výhoda tkví v jednoduchém nastavení zesílení prostřednictvím jediného rezistoru R9. Dále se zde využívá fakt, že se jedná o hotový rozdílový zesilovač, který není potřeba vytvářet z jednotlivých OZ, kde by bylo velmi obtížné dosáhnout tak vysoké vstupní symetrie. Takto předzesílené napětí putuje opět do usměrňovače složeného z OZ a (stejně tak jako v případě měření napětí) na analogový vstup Arduina.

4.6 Přepínání rozsahů

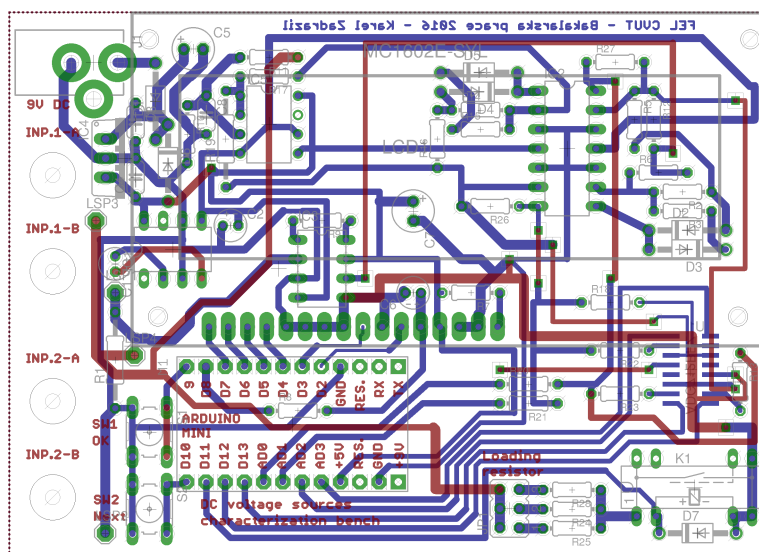
Jelikož měřené zdroje mají rozmanitá napětí a maximální proudy, bylo vhodné zapojení doplnit o volbu rozsahů. Vzhledem k osazení desky MCU je nejlepší volbou jejich přepínání plně automatizovat. Za tímto účelem mi byl vedoucím práce doporučen čtyřnásobný spínač ADG712BRZ. Má dobré vlastnosti v sepnutém stavu – odpor typicky $2,5 \Omega$. Jeho velkou nevýhodou je potom nemožnost použití pro spínání záporných napětí. Z toho důvodu nemohl být použit pro přepínání rozsahů přímo na AD620, jelikož tam se může objevovat jak kladné, tak záporné napětí. Jednou z variant bylo připojit ho k samostatnému OZ, který by se staral o další zesílení (případně i zeslabení) signálu. Toto řešení jsem zavrhl pro přeměnitelnou složitost – bylo by nutné přidat ještě další dva operační zesilovače. Při napájení z nulového napětí a zemnicím pinu připojeném na -5 V by byla splněna podmínka kladného napětí na spínaných vstupech. Vznikl by ale další problém týkající se řídicí logiky. Pro napětí -5V by byl spínač rozepnut a v 0V sepnut, což by nebylo kompatibilní s napěťovými úrovněmi Arduina, které jsou v tomto případě 0 a 5 V. Nakonec mě napadlo použít zapojení patrné ze schématu, tedy připojovat a odpojovat rezistory ve zpětné vazbě druhého operačního zesilovače. Na jeho výstupu totiž musí být vždy kladné napětí a na invertujícím vstupu je virtuální nula. Ani jedna strana spínače se tak nedostane pod nulovou úroveň napětí a nezapříčiní tak nepředvídané chování. Je zde zároveň zajištěno, že na drain těchto spínacích tranzistorů bude vždy přivedeno vyšší napětí, než na source, což je zřejmě pro správnou funkci spínače také potřeba.

4.7 DPS

Deska plošných spojů byla původně navrhovaná jako jednostranná, avšak během návrhu se ukázalo, že množství drátových propojek by bylo značné. Také kvůli dodržení předepsaných rozestupů a vzdálenosti spojů od okraje desky pro tento stupeň přesnosti výroby musely být některé další spoje přesunuty do vrchní vrstvy. Při návrhu jsem dbal především na to, aby spoje kterými mají protékat vyšší proudy, byly pro tento účel dostatečně dimenzované. Bohužel se kvůli nedostatku místa nepodařilo umístění relé a zatěžovacích rezistorů v těsné blízkosti měřicích svorek. Vzhledem k maximálnímu proudu 1 A však žádný větší úbytek napětí nehrozí. Stejně tak je tomu v případě rizika přehřátí rezistorů, jelikož tento proud bude téci vždy jen velmi krátkou dobu.

Problém by mohl nastat v případě relátka. To má jen velmi malou tepelnou kapacitu svých kontaktů, které by se mohly rychle přehřát a zdeformovat. Toto je ovšem ošetřeno v nahraném softwaru – pro potenciálně riziková napětí již měření se zvolenou nejnižší

hodnotou rezistoru neproběhne.



Obrázek 1: Návrh DPS vytvořený v programu Eagle

4.7.1 Zdířky

Všechny součástky použité v návrhu jsou použité z nějaké knihovny. S jedinou výjimkou a tou jsou zdířky pro připojení měřeného zdroje a zátěže. Jelikož se jedná pouze o otvor v jehož blízkosti se nachází pájecí ploška, nebylo těžké tento otvor čtyřikrát zreprodukovat přímo na DPS. Navíc se tím zjednodušil samotný návrh, jelikož místo připojení drátu nemusí být v knihovně přesně definované vzdálenosti, ale může se upůsobit aktuálním prostorovým dispozicím. Není totiž problém pro elektrické propojení konektoru s DPS použít delší či kratší kousek drátku.

4.7.2 Napájecí jack konektor

Pro tuto práci jsem modifikoval knihovnu pro napájecí jack konektor, kde jsem vypustil frézování plochých drážek pro osazení tohoto konektoru do DPS. Jelikož se deska vyráběla pouze v jediném exempláři, zbytečně by to prodražilo její výrobu. Aby se ale vývody konektoru daly vůbec prostrčit skrz DPS, bylo nutné zvětšit jejich průměr a tím i průměr pájecích plošek. Takto vzniklý prostor jsem potom vyplnil větším množstvím cínu. DPS je též doplněna o popisky ve vrstvě mědi. Toto řešení opět zjednodušilo její výrobu.

4.7.3 Osazení

Arduino a LCD displej jsem osadil do patice, aby mohl být jednoduše vyjmut a případně použit i v jiných aplikacích. To zároveň znamená i prostorovou výhodu, jelikož použitím vysokého konektoru vznikl pod displejem prostor, který se posléze dal využít k osazení většiny nižších součástek. Po předchozích zkušenostech z individuálního projektu, kdy použitý OZ LM324N vykazoval abnormální chování, jsem se rozhodl osadit do patic i všechny ostatní integrované obvody s drátovými vývody. Hledání závady je mnohem jednodušší, pokud je možnost samostatně otestovat obvod bez zapájeného integrovaného

obvodu, případně samotný integrovaný obvod v jiném zapojení na nepájivém poli. Zmíněný OZ v zapojení integrovaného stabilizátoru se i přes postupné vyvrácení všech možných externích příčin velmi přehříval. Žádný z výstupů nebyl proudově zatížen a taktéž jsem osciloskopem prověřoval, zda se žádný z obvodů nerozkmital. Nakonec jsem dospěl k závěru, že je samotný OZ vadný. Při napájení ± 8 V se obvod choval normálně, jen vykazoval mírně vyšší spotřebu. Nad tuto hodnotu jeho spotřeba se zvyšujícím se napětím exponenciálně rostla. Tento OZ by přitom měl pracovat až do napájecího napětí ± 16 V.

Vzhledem k těmto problémům s předchozím zapojením jsem se rozhodl celý řetězec otestovat na nepájivém poli. Zdálo se mi to rozumné především z důvodu větší komplexnosti celého zapojení. Testoval jsem celý řetězec od napěťového sledovače, až po integrovaný stabilizátor s proměnným ziskem. V případě CMOS spínačů ADG712 to bylo vyloženě nutné, jelikož se i přes dodržení všech předepsaných hodnot nejednalo přímo o doporučené zapojení. Abych se tedy vyhnul potencionálním komplikacím, zahrnul jsem do testování i tento obvod.

4.8 Program

V následující kapitole podrobně rozeberu jednotlivé části programu nahraného v Arduinu. Vlastní text programu je pro lepší přehlednost formátován dle zvyklostí vývojového prostředí Arduino IDE. Obsahuje velké množství komentářů. Ty mohou v budoucnu přispět ke snadnější orientaci při případné opravě.

4.8.1 Obsluha přístroje

Přístroj má dvě tlačítka, pomocí nichž jej lze obsluhovat. Tlačítko „Next“ slouží ke změně aktuální položky na displeji. Tlačítko OK potvrzuje volbu a na displeji se zobrazením náboje a energie je možné tyto dvě hodnoty dlouhým stiskem vynulovat.

4.8.2 Hlavička a knihovny

Každý program – ať už v jakémkoliv programovacím jazyce a hardware by měl na začátku v popisu obsahovat minimálně informaci o tom, o jaký program se jedná, jelikož název souboru nemusí být vždy dostatečně výmluvný a navíc ho uživatel může z nejrůznějších důvodů změnit. Následující program je napsaný v jazyce Wiring, ve kterém se Arduino programuje.

```
/*  
  Program pro zařízení na charakterizaci zdrojů stejnosměrného napětí  
  FEL ČVUT - Karel Zadražil  
  Bakalářská práce, 2016  
*/
```

Tato část importuje knihovny, které jsou potřeba pro běh samotného programu. První z nich slouží pro velmi jednoduchou obsluhu LCD displeje (bude rozebráno dále). Druhá v pořadí umožní vytvořit PWM na libovolném pinu. Toto řešení bylo nutné kvůli doplnění tlačítky na pinech 9 a 10. Ovládání kontrastu jsem přesunul na pin 8 s tím, že jsem si

neuvědomil, že nemá hardwarovou podporu PWM.

Poslední jmenovanou knihovnu je výhodné použít pro ukládání a načítání hodnot z vestavěné EEPROM paměti. Knihovny je potřeba importovat jednotlivě, jelikož každá knihovna zabírá určitý prostor programové paměti Arduina. Tento postup je podobný i jiným programovacím jazykům.

```
#include <LiquidCrystal.h> //importuje knihovny
#include <SoftPWM.h>
#include <EEPROM.h>
```

Zde si definujeme některé hodnoty, které jsou následně v kódu programu používány. Překladač na místo každé definované proměnné před vlastním přeložením dosadí její hodnotu. Tato sekce tedy nezabírá žádnou paměť navíc, jen usnadní změnu některých hodnot. Ty není nutné složitě dohledávat v kódu, ale dají se změnit zde, na jednom místě. Nehrozí tak, že by programátor na nějakou zapomněl, nebo omylem přepsal jinou.

```
//***** nastavení *****
#define kontrast      70      // nastavení kontrastu, (hodnota 0 -
                             255, nižší znamená tmavší)
#define kontrastpin   8       // kontrast ovládá pin 8
#define ok            9       // pin tlačítka OK (sestupná hrana
                             znamená stisk)
#define next          10      // pin tlačítka NEXT (sestupná hrana
                             znamená stisk)
#define rele          11      // relé
#define proud20       12      // piny kterými se ovládají příslušné
                             rezistory rozsahů, číslo
#define proud10       13      // značí jaký rezistor se jím sepne.
                             připnutím rezistorů v pořadí
#define napeti20      A3      // 20 a 10 se 2x a 4x zvýší rozsah
#define napeti10      A2
#define proudpin      A0      // vstup pro měření proudu
#define napetipin     A1      // vstup pro měření napětí
#define prahPrebuzeni 1015    // hodnota při které se přejde na vyšší
                             rozsah
#define prahPodbuzeni 400     // hodnota při které se přejde na nižší
                             rozsah
#define mereniOdporu  12000    // jak často se bude přeměřovat vnitřní
                             odpor zdroje [ms]
#define delkaMerOdporu 100     // jak dlouhý impuls se použije k měření
                             odporu [ms]
//*****
```

4.8.3 Deklarace globálních proměnných

Zde si deklarujeme globální proměnné, které budou dále v programu používány. Jejich hodnota je uložena v SRAM, a proto se stálým přepisováním čip neopotřebovává. Vzhledem k typu paměti jsou však data v nich uložená po vypnutí napájení ztracena. Proměnným je také možné již při deklaraci přiřadit hodnotu. Proměnné nazvané „napeti“, „proud“ atd. obsahují změřenou či vypočtenou hodnotu napětí, proudu atd. v aktuální vteřině. Tyto proměnné tedy slouží především k přímému výstupu na displej, nebo do

sériové linky. Proměnné „hodnotaProud“ a „hodnotaNapeti“ dočasně uchovává právě přečtenou hodnotu z analogového vstupu.

```
// vypočtené hodnoty
int hodnotaProud = 0; //[-] sem se budou ukládat hodnoty z měřicích pinů
                          k přepočtení (0-1023)

int hodnotaNapeti = 0; //[-]
float proud = 0; // [A] vypočtený proud z/do měřeného zdroje
float napeti = 0; // [V] vypočtené napětí zdroje
float napeti2 = 0; // [V] napětí se zatěžovacím rezistorem
float vykon = 0; // [W]
float energie = 0; // [Wh]
float nabojs = 0; // [Ah]
float odpor = 0; // [V]
float maxVykon = 0; // [V]
```

Zde jsou uchovávány konstanty, které byly vypočteny tak, jak napovídá komentář. Jedná se o teoretické hodnoty při použití dokonale přesných součástek. Kontrolním měřením jsem však zjistil, že takto získané hodnoty sedí s přesností na měřenou desetinu voltu a setinu ampéru. Dosažená přesnost je tak vyšší, než rozdíl mezi jednotlivými polaritami napětí a směry proudu. Je proto zbytečné snažit se je dále zpřesňovat.

Poté následují proměnné, jejichž komentář přímo v kódu je dostatečně jasný a není k tomu potřeba mnoho dodávat.

```
// konstanty sloužící jako dělitel pro napětí nebo proud
// zde je možné jednotlivé rozsahy kalibrovat
// delitelNapeti = rozsah_převodníku / napětová_referenc /
vstupní_odporový_dělič
// delitelProud = rozsah_převodníku * zesílení_AD620 /
napětová_referenc
const float delitelNapeti1x = 163.317 ; // = 1024/1.1/5.7
const float delitelNapeti2x = 81.6585 ; // .5* předchozí
const float delitelNapeti4x = 40.82925 ; // .5* předchozí
const float delitelProud1x = 3165.09 ; // = 1024*34/1.1
const float delitelProud2x = 1582.5454 ; // .5* předchozí
const float delitelProud4x = 791.27 ; // .5* předchozí

// proměnná, která se vždy po měření odporu inkrementuje na čas příštího
měření
unsigned long dalsiMereniOdporu = 0;
// proměnná která se vždy po měření inkrementuje o čas do dalšího měření
(1000 ms)
unsigned long cas = 0;

int rozsahU = 4; // kolikrát je aktuální rozsah vyšší než základní
int rozsahI = 4;
int disp = 1; // určuje který displej se má zobrazit
```

Do proměnné R rovnou ukládáme při její definici hodnotu uloženou v paměti EEPROM. Tuto paměť je naopak dobré přepisovat co nejméně, jelikož má výrobcem garantováno vysoké, ale omezené množství zápisů. Atmega328p by to dle katalogového listu měla vydržet 100 000 krát. V tomto projektu se do ní nikdy nezapisuje zcela automaticky, ale

pouze při ručním zásahu – při změně velikosti rezistoru v menu programu.

```
int R          = EEPROM.read(0); // načtení hodnoty zatěžovacího odporu z
EEPROM

// proměnné, do kterých se bude ukládat aktuální dělitel napětí/proudu
float delitelNapeti = delitelNapeti4x;
float delitelProud = delitelProud4x;
```

Za povšimnutí stojí definice znaku „Ω“ který jsem ve znakové sadě tohoto displeje nenašel. Proto jsem si definoval vlastní. Jedná se o pole proměnných typu byte – tedy proměnné, do kterých lze uložit jedničku nebo nulu. Jednička symbolizuje černý bod v matici displeje, nula naopak neztmavený segment. Velké písmeno „B“ na začátku každého řádku značí, že bude následovat binární hodnota.

```
// definice znaku omega pro displej
byte znak[8] = {B01110,
                B10001,
                B10001,
                B10001,
                B01010,
                B01010,
                B11011,
                B00000
               };
```

4.8.4 Inicializace a funkce „setup“

Ještě před vlastním použitím LCD displeje je potřeba provést jeden mezikrok. Vytvoření objektu, v mém případě nazvaném „LCD“. Zároveň s tím proběhne definice, na kterých pinech budeme s LCD komunikovat. Na tento objekt se bude program odvolávat vždy, když bude potřeba změnit obsah displeje.

```
// inicializuje LCD s definicí připojení na piny.
LiquidCrystal LCD(2, 3, 4, 5, 6, 7); // LCD(RS, Enable, D4, D5, D6, D7)
```

Poté už následuje funkce „setup“. Ta je v jazyce Wiring nezbytná a je jí nutné vytvořit i kdyby měla být prázdná. Všechny instrukce, které bude tato funkce obsahovat se provedou ihned po zapnutí napájení (případně resetu) a to pouze jednou.

Zde je použita k prvotnímu nastavení jednotlivých pinů do výchozí konfigurace pro další běh programu. Kvůli vyšší přesnosti referenčního napětí je použita interní napěťová reference 1,1 V. Je inicializované softwarové PWM, displej a sériová komunikace.

```
void setup() {
  pinMode(kontrastpin, OUTPUT); // nastavení pinů jako výstupy
  pinMode(ok, INPUT_PULLUP);    // a zapnutí pullup rezistorů u tlačítek
  pinMode(next, INPUT_PULLUP);
  pinMode(rele, OUTPUT);
  pinMode(proud20, OUTPUT);
  pinMode(proud10, OUTPUT);
  pinMode(napeti20, OUTPUT);
  pinMode(napeti10, OUTPUT);
}
```

```

digitalWrite(proud20, HIGH); // inicializace největšího rozsahu
digitalWrite(proud10, HIGH);
digitalWrite(napeti20, HIGH);
digitalWrite(napeti10, HIGH);

analogReference(INTERNAL); // přepne referenci na 1.1 V

SoftPWMBegin(); // inicializace softwarového PWM pro řízení kontrastu
LCD.begin(16, 2); // nastavení displeje na 16 znaků a 2 řádky
LCD.createChar(1, znak); // import znaku omega
Serial.begin(9600); // inicializace sériového výstupu
SoftPWMSet(kontrastpin, kontrast); // inicializace ovládání kontrastu

```

Na displej a do sériové linky se vypíše úvodní informace. Nejdříve si nastavíme kurzor do požadované polohy a poté vypíšeme hodnotu. Na konci displej smažeme, abychom na něj mohli vypisovat naměřené hodnoty.

```

// úvodní displej:
LCD.setCursor(0, 0);
LCD.print("Meric DC zdroju");
LCD.setCursor(0, 1);
LCD.print("DC source banch");

Serial.println("Meric stejnosmernych zdroju napeti");
Serial.println("DC voltage source test banch");
Serial.println("Karel Zadrazil");
Serial.println();
// legenda pro sériový výpis
Serial.println("napeti,proud,energie,naboj,vykon,[odpor]");

delay(2000);
cas = millis();
LCD.clear();
}

```

4.8.5 Funkce „loop“

Nejdůležitější funkcí je funkce loop. Jak již název napovídá, tato funkce běží neustále dokola a stejně jako funkce setup je i ona nedílnou součástí každého programu. Bez této funkce nepůjde zdrojový kód přeložit. Hned na začátku se testuje, zda-li nemá proběhnout další měření. Pokud ano, načtou se hodnoty z analogových vstupů.

```

void loop() {
  if ( millis() > cas ) { //pokud od posledního měření uběhl nastavený
                        čas, proběhne nové měření
                        // čtení hodnot
    hodnotaProud = analogRead(proudpin); // přečte hodnotu z A0
    hodnotaNapeti = analogRead(napetipin); // přečte hodnotu z A1

```

Poté se ověří, zda není potřeba změnit rozsah. Pokud ano, zavolá se příslušná funkce pro změnu rozsahu. Kontroluje se jak přebuzení tak i nedostatečné využití A/D převodníku. V proměnných „rozsahI“ a „rozsahU“ máme uloženy aktuální použité rozsahy, takže kontrolujeme i fakt, jestli je ještě kam zvyšovat nebo snižovat.

```

//kontrola rozsahu - pokud je mimo toleranci, proběhne změna rozsahu
if (hodnotaProud > prahPrebuzeni && rozsahI != 4 || hodnotaNapeti <
prahPodbuzeni && rozsahI != 1) rozsahProudu();
if (hodnotaNapeti > prahPrebuzeni && rozsahU != 4 || hodnotaNapeti <
prahPodbuzeni && rozsahU != 1) rozsahNapeti();

```

Po úspěšném odměření si můžeme nechat spočítat hodnoty napětí, proudu, výkonu, do této chvíle přenesené energie a náboje. Při případné změně rozsahu není potřeba znovu číst hodnoty napětí, nebo proudu, jelikož to už mají na starost tyto funkce samotné. Důležité je však nezapomenout inkrementovat proměnnou „cas“ o jednu sekundu. Za tuto dobu proběhne další měření. Pokud jsou splněny uvedené podmínky, bude se měřit i vnitřní odpor. Výsledky se vypíší na displej a do sériové linky příslušnými funkcemi.

```

// výpočty nových hodnot
napeti = hodnotaNapeti / delitelNapeti ;
proud = hodnotaProud / delitelProud ;
vykon = napeti * proud;
energie = energie + vykon / 3600;
naboj = naboj + proud / 3600;
cas = cas + 1000;

// měření odporu - proběhne pokud je tato hodnota aktuálně zobrazena
// a od posledního měření uplynul nastavený čas a zároveň je napětí
// zdroje dostatečné pro rozumné měření a zároveň bezpečné pro relé
if (disp == 3 && millis() > dalsiMereniOdporu && napeti > 0.2 &&
napeti / R < 1) {
    digitalWrite(rele, HIGH); // připne zatěžovací odpor
    delay(delkaMerOdporu); // počká než se hodnota ustálí
    napeti2 = analogRead(napetipin) / delitelNapeti; // přečte novou
                                                    hodnotu napětí

    digitalWrite(rele, LOW);
    dalsiMereniOdporu = millis() + mereniOdporu; // vypočte čas
                                                    případného dalšího měření

    odpor = (napeti - napeti2)/(napeti2 / R); // vypočte odpor zdroje
    maxVykon = napeti*napeti/R;
}

vypsati(); // výsledky vypíše po sériové lince
prekreslit(); // a překreslí displej
}

```

Funkci loop uzavírá část pro obsluhu tlačítek. Procesor není jinak příliš vytížen a proto můžeme tlačítka kontrolovat přímo v hlavní smyčce. Pokud by byl často zaneprázdněný nějakými časově náročnými operacemi, musely by se tlačítka obsluhovat prostřednictvím přerušení, aby uživatel nemusel tlačítko tisknout zbytečně dlouho. Velmi úsporným způsobem byly řešeny zákmity tlačítka. Předpokládá se, že zákmit nebude trvat déle než 20 milisekund. Proto se po zjištění stisku jednoduše provede vše potřebné a poté se počká ještě dalších 20 ms. Pak se testuje, zda-li již uživatel tlačítko uvolnil. Pokud ano, opět se počká dalších 20 ms a pokračuje se v běhu hlavní smyčky. To druhé čekání je nutné kvůli případným zákmitům při uvolňování tlačítka. Takový zákmit by se totiž projevil jako registrace a zpracování nového stisku.

Hlavní stavovou proměnnou této funkce je proměnná „disp“, která rozhoduje o tom, co se

bude zobrazovat na displej a zároveň jakým způsobem se budou chovat tlačítka.

```
// obsluha tlačítka "next" pokud je zobrazen některý hlavní displej
if (disp < 5) {
  if (digitalRead(next) == LOW) {
    if (disp == 4) disp = 1;    // pokud je zobrazen poslední (4.)
                                // displej, vrtit se na 1.
    else disp++;                // jinak inkrementovat
    prekreslit();
    delay(20);                  // zjednodušené ošetření
                                // zákmitů prodlevou
    while (digitalRead(next) == LOW) delay(1); // počká se na uvolnění
                                                // tlačítka
    delay(20);
  }
}
```

Pokud je aktuálně zobrazen displej 4 – tedy ten s informacemi pro vstup do menu výběru rezistoru, tlačítko OK změní proměnou „disp“ na 5, což je mód pro změnu hodnoty rezistoru.

```
// obsluha tlačítka "OK" pokud je zobrazen vstup do menu
// pokud je stisknuto, disp = 5 - tedy nastavení rezistoru
if (disp == 4) {
  if (digitalRead(ok) == LOW) {
    disp = 5;
    prekreslit();
    delay(20);
    while (digitalRead(ok) == LOW) delay(1);
    delay(20);
  }
}

// obsluha tlačítka "OK" pokud je zobrazen náboj a energie
// pokud je stisknuto víc než půl sekundy, pak se hodnoty vynulují
if (disp == 2) {
  int i = 0;
  while (digitalRead(ok) == LOW) {
    i++;
    if (i > 6) {
      naboj = 0;
      energie = 0;
      LCD.clear();
    }
    delay(100);
  }
  prekreslit();
}
```

Zde se obsluhují tlačítka pro displej s výběrem rezistoru. Poté, co je zvolena nová hodnota rezistoru se tento mód opustí (do proměnné „disp“ se uloží jednička). Ještě předtím se však nová hodnota zapíše do EEPROM, aby mohla být při startu znovu načtena a uživatel ji nemusel po každém odpojení napájení nebo resetu zadávat ručně znovu.


```

// obsluha tlačítek na displeji pro nastavení rezistoru
if (disp == 5) {
  // při stisku OK se hodnota R запиše do EEPROM a vrátíme se na
  displej č. 1
  if (digitalRead(ok) == LOW) {
    disp = 1;
    překreslit();
    EEPROM.write(0, R);
    delay(20);
    while (digitalRead(ok) == LOW) delay(1);
    delay(20);
  }
  // při stisku tlačítka next se změní hodnota proměnné R a překreslí
  displej
  if (digitalRead(next) == LOW) {
    switch (R) {
      case 15:
        R = 43;
        break;
      case 43:
        R = 100;
        break;
      default:
        R = 15;
        break;
    }
    překreslit();
    delay(20);
    while (digitalRead(next) == LOW) delay(1);
    delay(20);
  }
}
}
}

```

4.8.6 Funkce pro změnu rozsahu

Tato funkce je volána z hlavní smyčky v případě, že je potřeba změnit rozsah. Funkce pro změnu rozsahu napětí i proudu jsou téměř identické, proto si tu rozebereme jen tu pro proud. Rozdíly jsou pak čistě jen v názvech proměnných, metodika je u obou naprosto identická.

Funkce se hned na začátku větví podle toho, jaký rozsah je aktuálně zvolen. Pokud je jím ten nejvyšší, hned na začátku se zjistí, který se má použít a nastaví se odpojení příslušných rezistorů.

```

// funkce pro automatickou změnu rozsahu proudu
void rozsahProudu() {
  switch (rozsahI) {
    case 4:
      // pokud byl rozsah 4
      if (hodnotaProud < prahPodbuzeni / 2 ) { // a aktuální hodnota je
        // méně jak polovina prahové
        rozsahI = 1; // pak se zvolí rozsah 1
        delitelProud = delitelProud1x; // a dle toho se upraví i
        // nastavený dělitel
        digitalWrite(proud20, LOW); // fyzicky se nastaví
        // příslušný rozsah
        digitalWrite(proud10, LOW);
      }
  }
}

```

```

else if (hodnotaProud < prahPodbuzeni) { // pokud je menší než práh
                                     // ale nesplnil předchozí podmínku
                                     // změní se rozsah na 2
    rozsahI = 2;
    delitelProud = delitelProud2x;
    digitalWrite(proud20, HIGH);
    digitalWrite(proud10, LOW);
}
break;

```

Pokud je aktivní prostřední rozsah, tak se aktuální rozsah zvýší či sníží podle potřeby.

```

case 2: // pokud byl rozsah 2
    if (hodnotaProud > prahPrebuzeni) { // zjistí se jestli se má
                                        // rozsah zvýšit
        rozsahI = 4;
        delitelProud = delitelProud4x;
        digitalWrite(proud20, HIGH);
        digitalWrite(proud10, HIGH);
    }
    else if (hodnotaProud < prahPodbuzeni) { // nebo snížit
        rozsahI = 1;
        delitelProud = delitelProud1x;
        digitalWrite(proud20, LOW);
        digitalWrite(proud10, LOW);
    }
    break;

```

Poslední případem už nemůže být jiný, než nejnižší rozsah. Ostatní případy jsme si již podchytili a ošetřili. Tato část programu se vykonává, pokud je zjištěno, že je vstup přebuzený. Není jisté jak hodně, na který z vyšších rozsahů je potřeba přepnout. Proto zvýšíme rozsah jen o jeden stupeň. Poté počkáme 60 ms na ustálení stavu. Znovu změříme přicházející napětí. Pokud je v normě, pak už rozsah neměníme. Pokud není, zvýšíme na nejvyšší a můžeme funkci pro změnu rozsahu opustit.

```

default: // pokud byl rozsah 1, byl
         // přebuzený
         // zvýší se na 2
    rozsahI = 2;
    delitelProud = delitelProud2x;
    digitalWrite(proud20, HIGH);
    digitalWrite(proud10, LOW);
    delay(60); // počká se na ustálení stavu
    hodnotaProud = analogRead(proudpin); // a znovu se přečte hodnota
                                        // napětí
    if (hodnotaProud > prahPrebuzeni) { // pokud je vstup i přesto
                                        // přebuzený
                                        // zvýší se rozsah na 4
        rozsahI = 4;
        delitelProud = delitelProud4x;
        digitalWrite(proud20, HIGH);
        digitalWrite(proud10, HIGH);
    }
    break;
}
delay(60); // po změně rozsahu se počká
           // na ustálený stav
hodnotaProud = analogRead(proudpin); // a znovu se přečte hodnota
                                     // proudu
}

```

4.8.7 Funkce pro přepisování displeje

Toto je funkce, která se periodicky volá vždy, když je potřeba aktualizovat data na displeji. Funkce se skládá z několika částí. Vždy se vykoná pouze jedna, v závislosti na stavu proměnné „disp“. Touto proměnou je totiž vybrán konkrétní stav, který je potřeba vypsát na displej.

```
// funkce pro přepsání displeje
// v závislosti na tom, v jakém módu se přístroj nachází
// mezery navíc se vypisují kvůli případnému přepsání informace
// z předchozího displeje
void prekreslit() {
    // DISP 1
    if (disp == 1) {
        LCD.setCursor(0, 0);
        LCD.print("Napeti ");
        LCD.print(napeti, 3);
        LCD.setCursor(13, 0);
        LCD.print(" V ");
        LCD.setCursor(0, 1);
        LCD.print("Proud ");
        LCD.print(proud, 3);
        LCD.setCursor(13, 1);
        LCD.print(" A ");
    }

    // DISP 2
    if (disp == 2) {
        LCD.setCursor(0, 0);
        LCD.print("Energie ");
        LCD.print(energie, 4);
        LCD.setCursor(14, 0);
        LCD.print("Wh");
        LCD.setCursor(0, 1);
        LCD.print("Naboj ");
        LCD.print(naboj, 4);
        LCD.setCursor(14, 1);
        LCD.print("Ah");
    }

    // DISP 3
    if (disp == 3) {
        LCD.setCursor(0, 0);
        LCD.print("Vykon ");
        LCD.print(vykon, 3);
        LCD.setCursor(14, 0);
        LCD.print(" W ");

        LCD.setCursor(0, 1);
        LCD.print("Odpor ");
        LCD.print(odpor, 3);
        LCD.setCursor(14, 1);
        LCD.print(" ");
        LCD.write(1); // můj znak omega
        LCD.print(" ");
    }
}
```

```

// DISP 4
if (disp == 4) {
  LCD.setCursor(0, 0);
  LCD.clear();
  LCD.print("Vstoupit do");
  LCD.setCursor(0, 1);
  LCD.print("nastaveni ");
}

// DISP 5
if (disp == 5) {
  LCD.setCursor(0, 0);
  LCD.clear();
  LCD.print("Hodnota rezistoru:");
  LCD.setCursor(0, 1);
  LCD.print(R);
  LCD.setCursor(4, 1);
  LCD.write(1);      // můj znak omega
}
}

```

4.8.8 Funkce pro výpis do sériové linky

Toto je definice funkce, která ovládá výpis do sériové linky. Jednotlivé hodnoty jsou odděleny čárkou, takže je vhodné zachycenou komunikaci rovnou ukládat do *.csv souboru, který je dále zpracovatelný. Program vypisuje všechny vypočtené hodnoty a pokud byl změřen vnitřní odpor, tak vypíše i jej. Perioda měření vnitřního odporu je sice stálá, avšak měří se pouze v případě, že jsou splněny další podmínky. Proto jsem se rozhodl vypisovat hodnotu vnitřního odporu pouze v případě nového měření. Nebylo by totiž poznat, která hodnota je nová a která jen kopíruje tu předchozí.

```

// vypisuje naměřené hodnoty přes sériovou linku v tomto pořadí:
// napeti, proud, energie, naboj, vykon, [odpor]
// legendu vypíše funkce setup() při zapnutí napájení či po resetu
void vypsati() {

  Serial.print(napeti, 5);
  Serial.print(",");
  Serial.print(proud, 5);
  Serial.print(",");
  Serial.print(energie, 5);
  Serial.print(",");
  Serial.print(naboj, 5);
  Serial.print(",");
  Serial.print(vykon, 5);
  //pokud měření neproběhlo během poslední půl vteřiny, tak jen odřádkuj
  if (dalsiMereniOdporu + 500 < millis() + mereniOdporu)
Serial.println();
  //jinak se vypíše také odpor
  else {
    Serial.print(",");
    Serial.println(odpor, 5);
  }
}
}

```

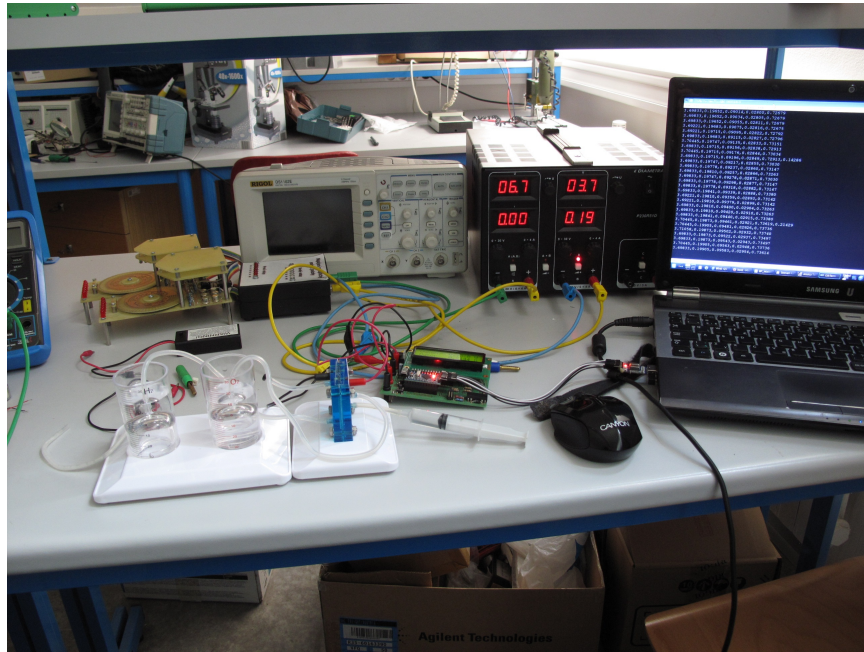
4.8.9 Kompilace a flashování

Celý program včetně knihoven zabírá po kompilaci zhruba 9,5 Kb, což je třetina využitelného místa pro program. Nelze totiž využít celé vnitřní paměti Atmegy, část už totiž zabírá bootloader sloužící k zapisování nového programu. Nahrání se provádí prostřednictvím UART rozhraní. Jelikož nejpohodlnější možnost jak takový program ladit je jeho nahrání do Atmegy a sledování jeho chování v reálném zapojení, bylo nutné mnohokrát do Arduina nahrát nový program. To se provádí tak, že se Arduino resetuje a poté asi půl sekundy čeká na nahrání nového programu. Pokud se tak nestane, začne vykonávat ten již uložený.

Atmegu je tedy nutné těsně před nahráním nového programu resetovat. To se dá udělat ručně, podržením fyzického tlačítka „reset“ a uvolněním ve správný okamžik. Toto řešení však není pohodlné a vyžaduje i správný odhad. Proto jsem se rozhodl si ze svého převodníku vyvést pin DTR_N který Arduino IDE před nahráním programu podrží v nule. Pokud ho tedy spojíme s pinem RST na Arduinu, resetuje se ve správný čas zcela automaticky. Avšak vzhledem k tomu, že na DTR_N je nula během celého nahrávání programu, při přímém propojení by se bootloader vůbec nerozběhl. Proto jsou piny propojené skrz kondenzátor 0.1 μ F, který při sestupné s sebou stáhne i napětí na RST pinu držené pullup rezistorem v jedničce, ale po nabití kondenzátoru se hodnota napětí na RST pinu vrátí zpět na úroveň logické jedničky. Spustí se bootloader čekající na přenos a nahrávání programu již nic nebrání. Fotografie upraveného převodníku naleznete v příloze II.

5 Testování

5.1 Měření s vodíkovým článkem



Obrázek 2: Pohled na pracoviště při testování na vodíkovém článku

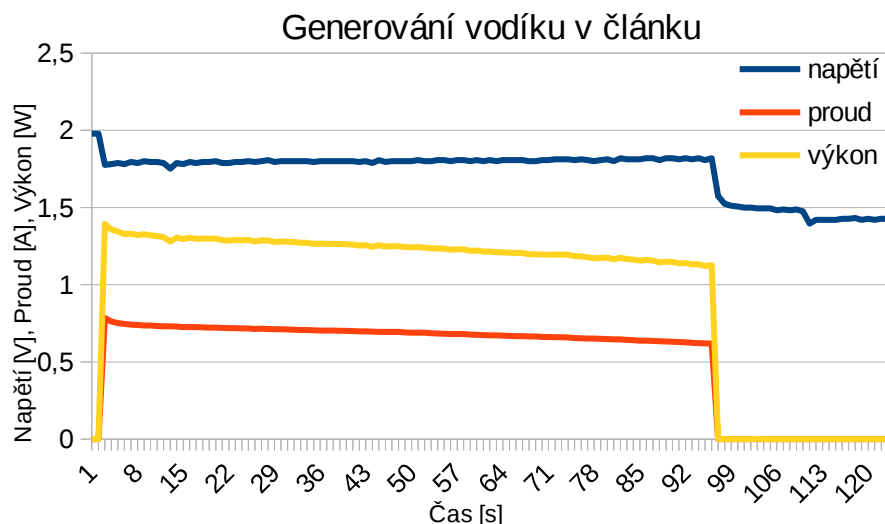
Během testování s vodíkovým palivovým článkem se přístroj choval zcela dle očekávání. Zkoušel jsem ho v módu generace vodíku i výroby elektřiny. Zde je možné si prohlédnout ukázkou výstupu:

```
Merici stejnosměrných zdrojů napětí  
DC voltage source test bench  
Karel Zadrazil
```

```
napeti, proud, energie, naboj, vykon, [odpor]  
1.97775,0.00095,0.00000,0.00000,0.00187  
1.97775,0.00032,0.00000,0.00000,0.00062  
1.77569,0.78481,0.00039,0.00022,1.39359  
1.78181,0.76207,0.00076,0.00043,1.35786  
1.78793,0.75196,0.00114,0.00064,1.34445  
1.78181,0.74564,0.00151,0.00085,1.32858  
1.79406,0.74185,0.00188,0.00105,1.33091  
1.78793,0.73932,0.00224,0.00126,1.32185  
1.80018,0.73679,0.00261,0.00146,1.32635  
1.79406,0.73553,0.00298,0.00167,1.31958  
1.79406,0.73300,0.00334,0.00187,1.31504  
1.78793,0.73174,0.00371,0.00207,1.30829  
1.75120,0.73047,0.00406,0.00228,1.27920  
1.78793,0.72921,0.00443,0.00248,1.30377
```

1.78181, 0.72668, 0.00479, 0.00268, 1.29481
1.79406, 0.72668, 0.00515, 0.00288, 1.30370, 0.10309

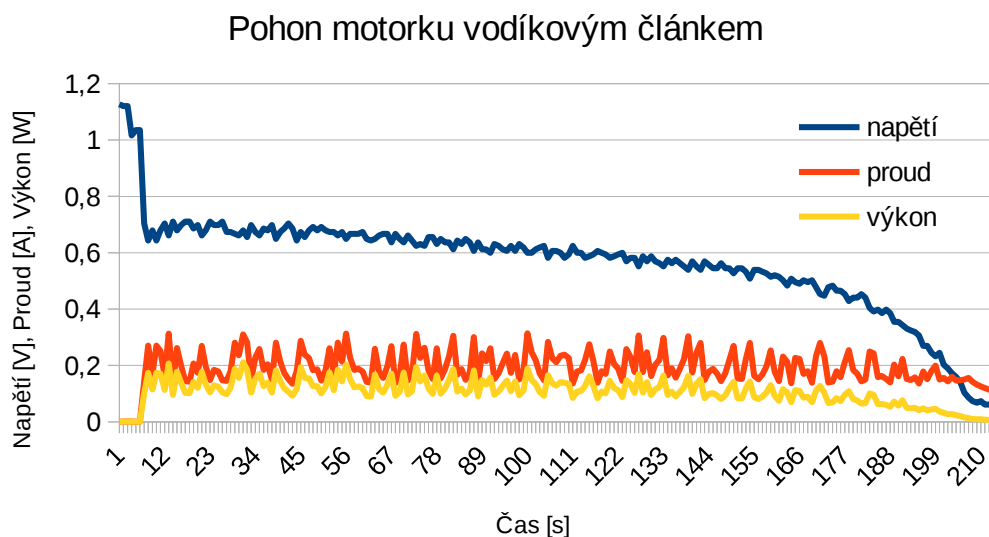
Tyto hodnoty byly získány při elektrolýze vody. Jak je vidět, tak při konstantním napětí (2,4 V) se proud článkem generující vodík a kyslík postupně zmenšoval spolu s tím, jak se voda rozkládala na své plynné složky. Níže je graf, který odpovídá předešlému výpisu.



Graf 1: Průběh proudu a napětí při generování vodíku

Během druhé vteřiny byl připojen palivový článek ke zdroji. V 97. byl zdroj napětí odpojen. Po 111 vteřinách došlo k měření vnitřního odporu, což se projevilo skokovým snížením napětí i přesto, že nebyl z článku odebírán žádný proud. K vyrobení 20 ml vodíku bylo použito 32,3 mWh.

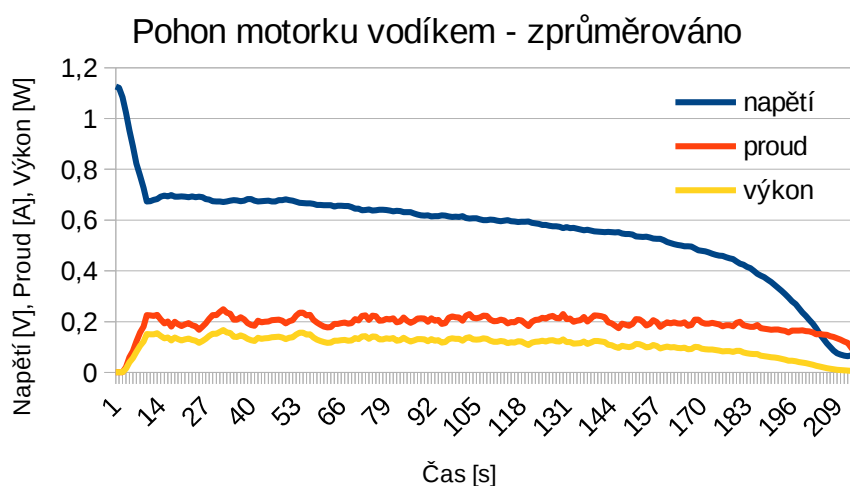
Takto už vypadá graf pro vodíkový článek jako generátor elektrického proudu:



Graf 2: Graf pro generování elektrické energie v článku

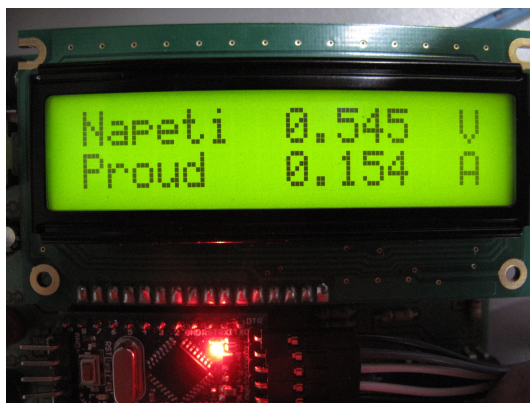
Jako zátěž byl použit elektrický motorek dodávaný spolu se soupravou článku. Jelikož byly vzorky napětí a proudu získávány z velmi krátkého časového intervalu, projevila se zde časová nestálost odběru proudu motorku. Velmi totiž záleží na tom, v jaké poloze se nacházel rotor motorku v době odebrání příslušného vzorku.

Jelikož je vzorkovací frekvence relativně vysoká, možné hodnoty vyhladit například metodou klouzavého průměrování.



Graf 3: Klouzavý průměr grafu generovaného proudu a napětí

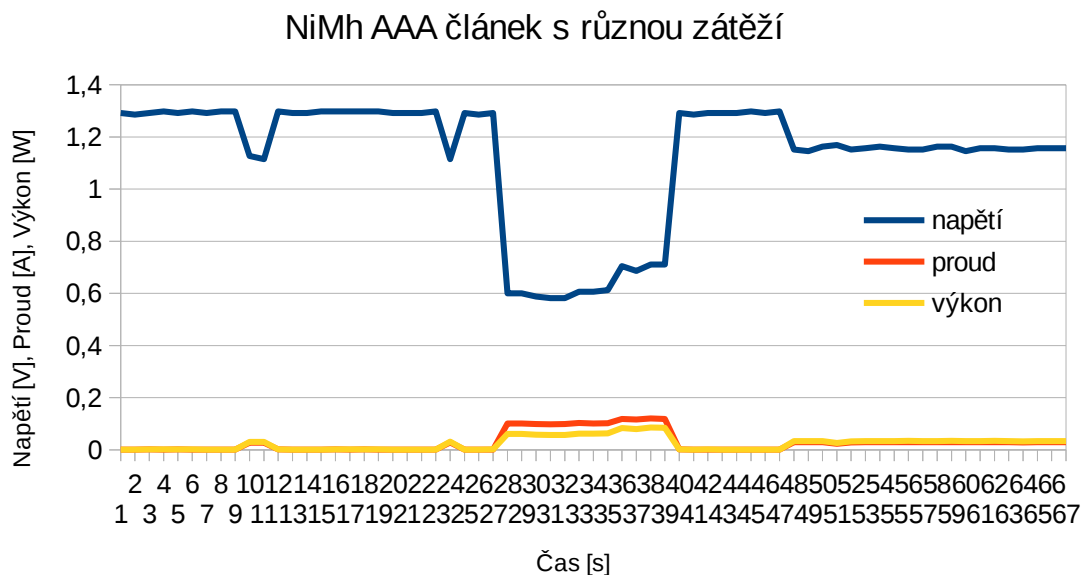
Vnitřní odpor se po celou dobu měření pohyboval kolem hodnoty 1,5 Ω .



Obrázek 3: Zobrazení napětí a proudu na displeji

5.2 NiMH články

Testování jsem opakoval také na niklo-metalhydridových mikrotužkových bateriích s původní kapacitou 700 mAh. Výstup z tohoto měření naleznete v grafu 4.



Graf 4: Zachycená charakteristika NiMH článku

Jak je vidět, odebíraný proud opět koresponduje se změnou napětí. Tentokrát je úbytek na vnitřním odporu mnohem výraznější, čemuž odpovídá i naměřený vnitřní odpor 6Ω bez zátěže.



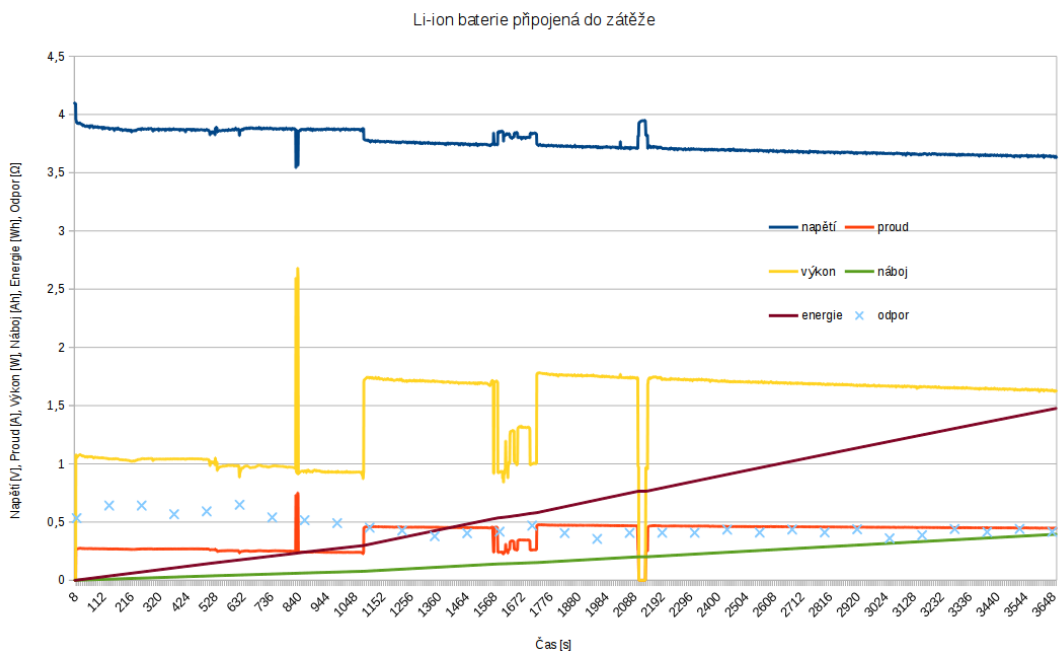
Obrázek 4: Použité NiMH články

5.3 Zatěžovací rezistory

Pro zatížení zdroje definovaným proudem jsou použity rezistory R24, R25 a R28 přepínatelné jumperem. O změně zatěžovacího rezistoru se bohužel Arduino samo nemá jak dozvědět, a proto je nutné provedenou změnu také pomocí ovládacích tlačítek oznámit mikrokontroléru. Jelikož se jedná o poměrně velké proudy a zároveň polarita připojeného zdroje může být libovolná, rozhodl jsem se ke spínání použít místo MOSFET tranzistoru jazýčkové relé, které za příznivou cenu zvládne spínat proudy do 1 A a časy přitahu a odpadu (0,5 ms; 0,2 ms) [9] zde nejsou limitujícím faktorem. Díky nízkému napětí cívky 5 V a proudu 10 mA bylzeměnění možno toto relé připojit přímo na digitální výstup Arduina. Zapojení bylo doplněno pouze o ochrannou diodu D7. Ta má zabránit zničení

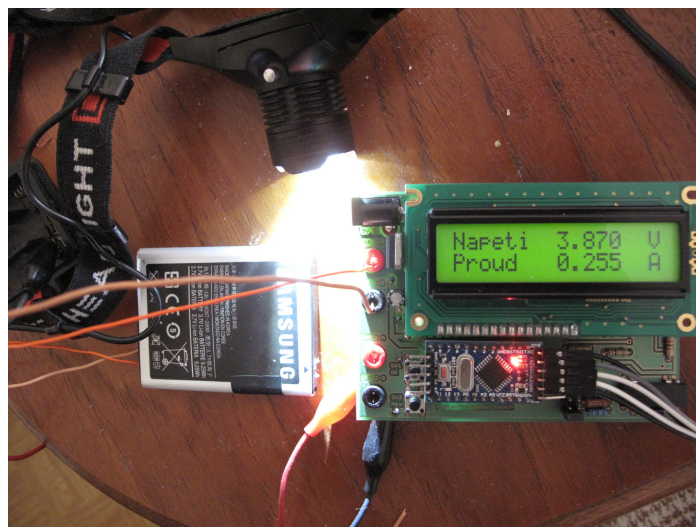
výstupu napětíovou špičkou, která vznikne v okamžiku odpojení cívky. Dioda se za běžného provozu neprojeví, jelikož je orientovaná závěrně. V případě rychlého vypnutí se na cívce naindukuje vysoké napětí opačné polarity, než bylo přítomné budící. To by mohlo mít pro připojený mikrokontrolér fatální následky, avšak dioda se v tomto případě otevře a napětíovou špičku omezí.

5.4 Měření na Li-ion článku



Graf 5: Experiment s vybíjením Li-ion

Pro účely testování jsem záměrně vybral velmi opotřebovanou baterii, která již není v původním zařízení použitelná z důvodu vyššího vnitřního odporu a malé kapacity. Díky tomu je na grafu dobře patrné, jak se mění napětí při experimentování se zátěží. Z grafu je vidět, jak zatížení baterie koresponduje s jejím napětím.

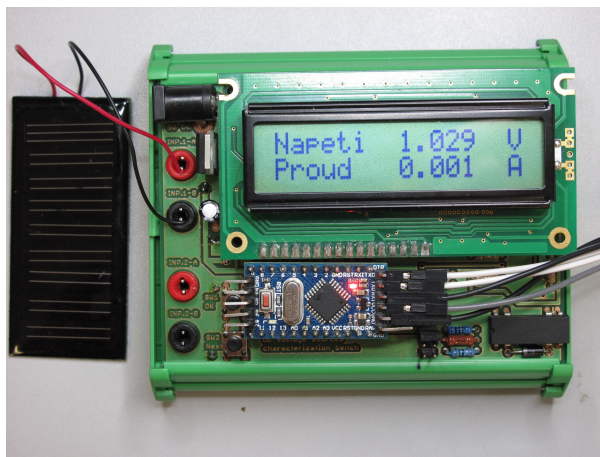


Obrázek 5: Testování na Li-ion baterii a reálném spotřebiči – LED svítilně

Zde se již pomalu začíná narážet na limity LibreOffice, vykreslování grafu již bylo poměrně náročnou záležitostí.

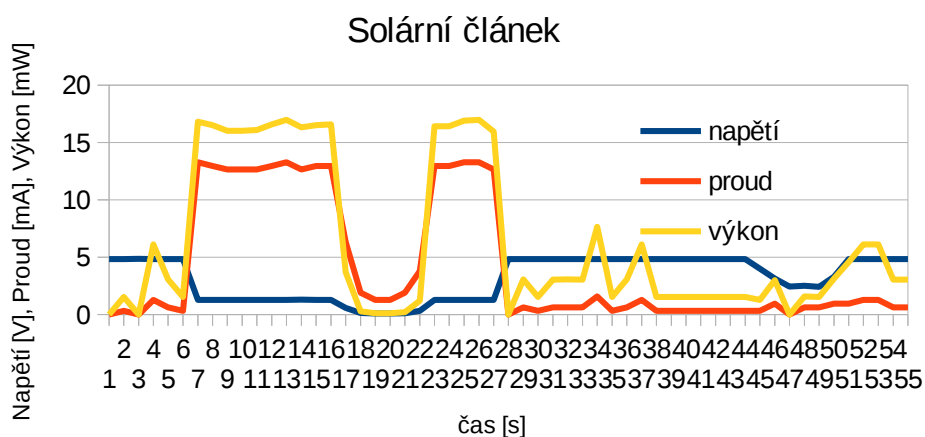
Při pozorování vývoje vnitřního odporu v čase je patrné, že zatížený li-ion článek se chová jako tvrdší zdroj (byť nižšího napětí), než článek bez zátěže.

5.5 Solární článek



Obrázek 6: Testování na solárním článku

Posledním testovaným zdrojem byl solární článek. Opět jsem zachycenou komunikaci zpracoval a vynesl do grafu.



Graf 6: Výstup testování na solárním článku

Na začátku je článek nezatížený, v 6. vteřině jsem připojil zatěžovací rezistor. Mezi 12. a 23 vteřinou jsem článek zaclonil a mezi 44. a 50. vteřinou je pozorovatelná reakce na postupné zaclonění článku naprázdno. Ojedinéle proudové peaky jsou způsobené šumem operačních zesilovačů.

6 Možnosti dalšího vylepšení

Jako asi každý projekt, tak i tento má mnoho míst, která by se dala vylepšit. Některé záležitosti je možné vyřešit pouhým přepsáním kódu programu, jeho zkompilováním a nahráním do modulu Arduina. Jiné by si ovšem vyžádala změny na DPS.

6.1.1 Detekce polarity napětí a směru proudu

Schéma by bylo možné doplnit o další komparátor, který by porovnával vstupní napětí usměrňovače s nulovým napětím. Získaný signál by potom sloužil jako informace o polaritě přiloženého napětí. Arduino by pak mělo informaci, kterým směrem proud teče a s jakou polaritou je zapojen zdroj napětí. Předpokládám však, že tuto informaci obsluha stejně má. Ví, s jakou polaritou připojila zdroj a stejně tak ví, kterým směrem má očekávat procházející proud.

6.1.2 Elektroměr

Zařízení by bylo také možné s úpravou software a po doplnění o měřicí transformátor použít i jako velmi citlivý elektroměr. Většina komerčně nabízených elektroměrů totiž neregistruje odběry do 1 W, u tohoto by to nebyl žádný problém.

6.1.3 Konektivita a záznam

Další možností jak rozšířit funkcionalitu tohoto přístroje je například připojení SD karty, kam by se zaznamenávala naměřená data. Vnitřní paměť je pro delší záznam příliš malá, avšak i nejmenší SD karty poskytnou dostatečnou kapacitu pro mnoho hodin až dní záznamu. SD karta však obsadí piny využívající SPI, konkrétně 11, 12 a 13. Vzhledem k tomu, že všechny základní piny jsou použity pro jiné účely, bylo by nutné přepracovat DPS a celé řešení přizpůsobit pro jeden konkrétní klon Arduina Pro Mini. SD karta pracuje s napájecím a signálovým napětím 3,3 V a není 5 V kompatibilní. Bylo by tedy potřeba vyřešit napájení a případný převod napětíových úrovní.

Jinou možností připojení by byl například wi-fi či ethernetový modul. Toto řešení by opět vyřešilo galvanické oddělení citlivé elektroniky počítače od měřicího přístroje a zároveň autonomní záznam, pomíneme-li běžící server, který bude data zaznamenávat. Levné wi-fi moduly s čipem ESP8266 však pracují též na napětí 3,3 V. Proto by stejně jako v případě připojení SD karty bylo nutné počítat s převodníkem. Pak se však využijí i piny RX a TX pro sériovou linku, které se používají pouze k programování a nebo k přenosu dat do PC. Zde by tedy nebyly potřeba.

6.1.4 Jas

V případě dostatku volných pinů by mohl být jas displeje ovládaný prostřednictvím programu. Stačil by k tomu pouze jeden tranzistor zapojený jako spínač a předřadný rezistor do jeho báze. Piny Arduina totiž nejsou konstruované na větší proud než 20 mA.

6.1.5 Kontrast

Kontrast by bylo lepší řídit pomocí hardwarového PWM pinu, jelikož softwarový PWM zbytečně zatěžuje procesor. Na druhou stranu program hlavní smyčky není tak náročný, aby toto řešení znamenalo nějaké postřehnutelné zpomalení jeho běhu. Pokud by však byl použit náročnější program, kde by výkon procesoru mohl být limitující, pak by s použitím softwarového mohl být problém. Jinou (a také zvažovanou) možností by bylo použití odporového trimru. Otázkou však zůstává, zdali by to bylo pro obsluhu přístroje pohodlnější.

7 Závěr

Zkonstruovaný přístroj se chová dle teoretických předpokladů a je dobře použitelný k běžnému měření stejnosměrných zdrojů napětí v jeho pracovním rozsahu. Nepřináší příliš vysokou zátěž pro měřený obvod a je vhodný i k dlouhodobému měření. Naopak není určeno pro měření napětí menších, než je desetina voltu a proudů menších než 5 mA.

Zařízení je i přes převážné nasazení THT součástek kompaktní a dobře se s ním pracuje. Jako jediný neduh týkající se uživatelské přívětivosti vidím umístění potvrzovacího tlačítka, které díky vysoké patici Arduina není při plném obsazení druhého vstupu rychle přístupné.

Běh programu je stabilní a svižný. Některé funkce jsou v programu řešeny zbytečně složitě, jelikož kód jako takový vznikl přidáváním dalších funkcí k již stávajícímu základu. Tento styl psaní se zjevně neosvědčil.

Po připojení do měřeného obvodu svými parametry nenarušuje výrazným způsobem tok proudu měřeným obvodem, což byl ostatně jeden z cílů této práce. Měřený obvod může fungovat s nezměněnými parametry.

Také se podařilo do zapojení zakomponovat přepínání rozsahů, které funguje bez jakéhokoliv zásahu uživatele.

7.1 Technické údaje:

Napájení adaptérem:	8 – 12V
Napájení prostřednictvím USB:	5V
Spotřeba:	max 150 mA
Maximální rozsah měřeného napětí:	± 25 V
Další napěťové rozsahy	$\pm 12,5$ V, ± 6 V
Maximální rozsah měřeného proudu:	$\pm 1,3$ A
Další proudové rozsahy	± 650 mA, ± 650 mA, ± 325 mA
Vstupní odpor pro měření napětí:	> 4 M Ω
Vložený měřicí odpor:	0,1 Ω
Maximální přípustná hodnota proud:	$\pm 5,48$ A
Maximální přípustná hodnota napětí:	± 30 V

7.2 Poznámky k použití

Vzhledem k tomu, že byla celá konstrukce navrhována s ohledem na maximální jednoduchost použití, stačí pouze připojit napájecí napětí 8 až 12V a automaticky dojde k zahájení měření. Na displeji se zobrazuje aktuální napětí, proud, výkon nebo vnitřní odpor zdroje. Nulování je možné provést červeným tlačítkem reset přímo na desce

Arduino, případně krátkým odpojením napájení.

Jak již bylo zmíněno, připojení zdroje a jeho polarita je libovolná. Lze využít horní nebo spodní pár konektorů (INP.1 nebo INP.2), přičemž do zbývajících páru je možno připojit zátěž. Červená a černá barva indikuje korespondující si zdířky.

Další využití skýtá výpis hodnot po sériové lince, který po zachycení lze dále zpracovávat.

Seznam použité literatury

- [1] Josef Sedlák. Řešení elektronických obvodů. Zesilovače. [online]. 23.5.2003 [cit. 2016-05-03]. Dostupné z: <http://www.zesilovace.cz/rservice.php?akce=tisk&cislocclanku=2003052302>
- [2] Václav Nývlt. Každá baterie jde dobít, i ta nenabíjecí. Speciální nabíječkou. Technet.cz. [online]. 5. 10. 2010 [cit. 2016-05-16]. Dostupné z: http://technet.idnes.cz/test-kazda-baterie-jde-dobit-i-ta-nenabijeci-specialni-nabijeckou-pxi-/tec_technika.aspx?c=A101004_080940_tec_technika_nyv
- [3] Elektrotechnika 1 – SS zdroje [online]. 2011. [cit. 2016-05-16]. Dostupné z: https://www.fd.cvut.cz/personal/janes/elektrotechnika1/prednasky prezentace/predn08az11_ss_zdroje.pdf
- [4] Bc. Roman Losenický. Diplomová práce – Nabíječ akumulátorů s mikrokontrolérem [online]. 2011. [cit. 2016-05-16]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=37452
- [5] Atmel. Katalogový list Atmega328p [online]. 2015. [cit. 2016-05-17]. Dostupné z: http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf
- [6] Ovládání znakových LCD s řadičem HD44780. Pandatron.cz: Elektrotechnický magazín. [online]. 17.3.2009 [cit. 2016-05-17]. Dostupné z: http://pandatron.cz/?685&ovladani_znakovych_lcd_s_radice_m_hd44780_%96_1._dil
- [7] Analog Devices. Katalogový list AD620 [online]. 2003. [cit. 2016-05-16]. Dostupné z: <http://www.analog.com/media/en/technical-documentation/data-sheets/AD620.pdf>
- [8] Using current limiting resistors on AVR I/O pins. ScienceProg. [online]. 16.3.2014 [cit. 2016-03-05]. Dostupné z: <http://www.scienceprog.com/using-current-limiting-resistors-on-avr-io-pins/>
- [9] COSMO. Katalogový list D1A051000 [online]. 1998. [cit. 2016-05-17]. Dostupné z: www.cosmo-ic.com/object/products/D1A051000.pdf

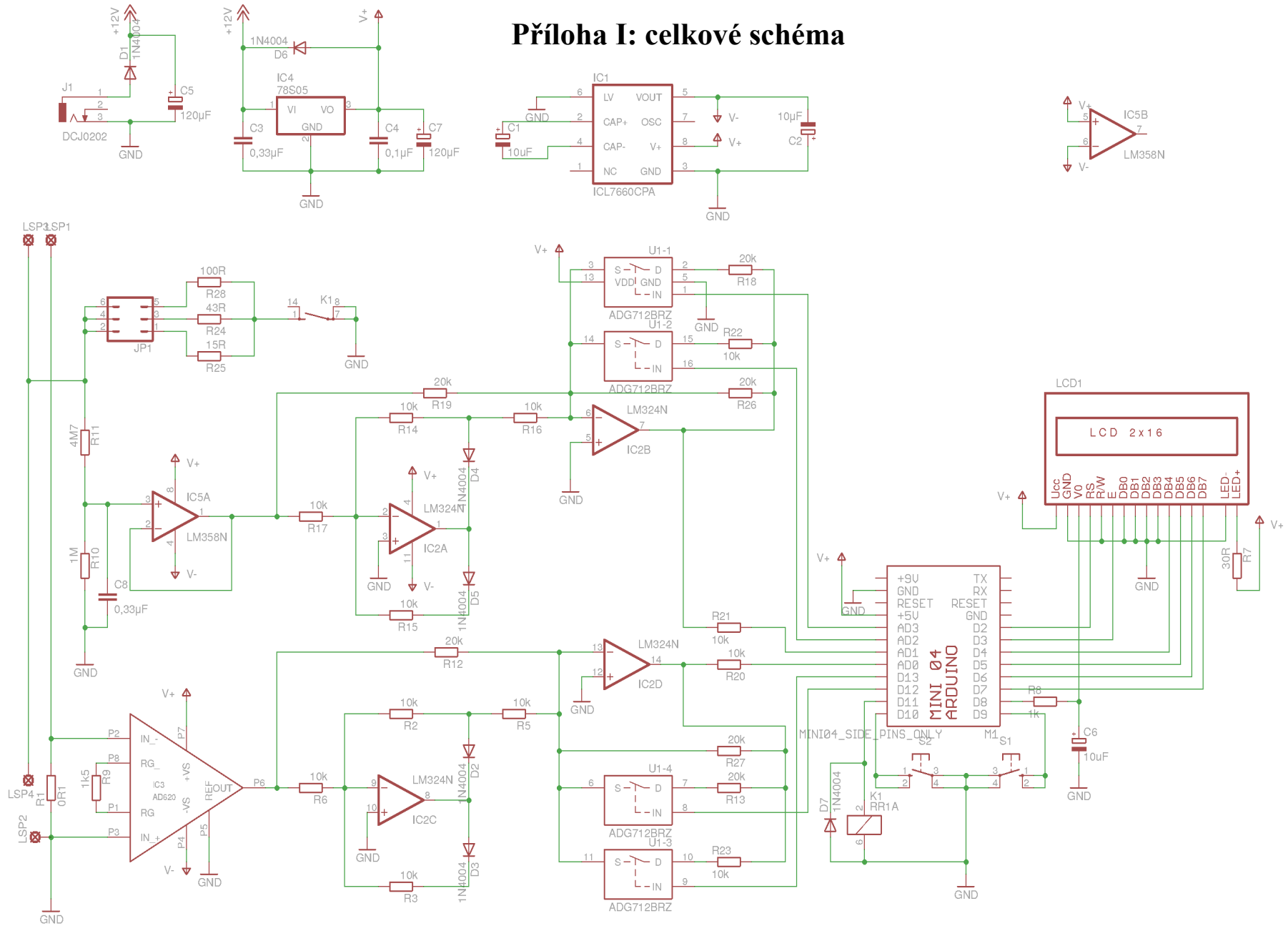
Seznam příloh:

Příloha I	celkové schéma
Příloha II	fotografie

Seznam příloh na CD:

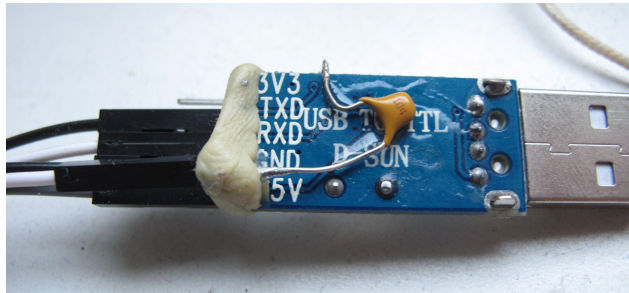
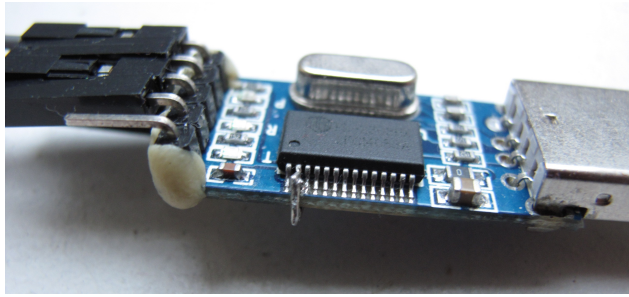
bakalarska_prace.pdf	celý text této bakalářské práce
program.ino	zdrojový kód programu
zarizeni.brd	návrh desky pro eagle
zarizeni.sch	schéma pro eagle
generator_vodiku.log	log z testování palivového článku jako generátoru plynu
solarni_clanek.log	log pro solární článek
spotreba_vodiku.log	log testování výroby energie v palivovém článku
Li-ion_test.log	test s li-ion baterií
NiMH_vybijeni.log	test s NiMH baterií

Příloha I: celkové schéma

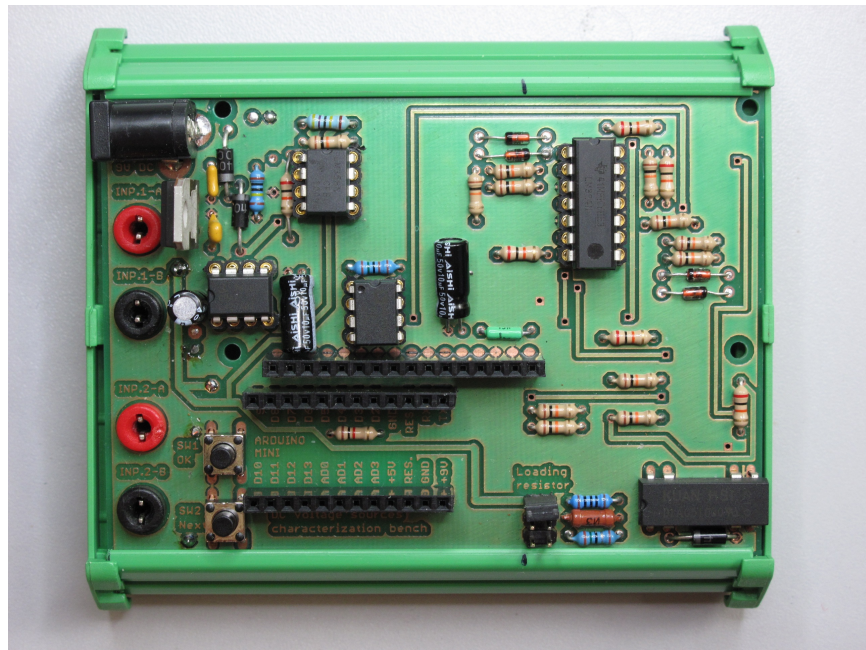


Příloha 2 - fotografie

Upravený převodník



Pohled na DPS ze svrchní strany bez osazeného Arduina a displeje



Pohled na spodní (vnitřní) stranu

