

Univerzita Pardubice  
Dopravní fakulta Jana Pernera

System pro záznam dat z fotovoltaického panelu  
Jakub Švadlenka

Bakalářská práce

2017

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jakub Švadlenka**  
Osobní číslo: ..... **D13934** .....  
Studijní program: **B3709 Dopravní technologie a spoje**  
Studijní obor: ..... **Elektrotechnické a elektronické systémy v dopravě**  
Název tématu: **Systém pro záznam dat z fotovoltaického panelu**  
Zadávací katedra: **Katedra elektrotechniky, elektroniky a zabezpečovací  
techniky v dopravě**

### Z á s a d y p r o v y p r a c o v á n í :

Předmětem práce je navrhnout a realizovat automatický měřicí systém se záznamem pro podporu statistického zpracování energetických zisků z 8 nezávislých fotovoltaických panelů (nezávislý záznam U a I pro každý panel, teplotu panelu a okolí).

Měřicí systém musí být schopen provádět bezobslužný záznam všech sledovaných veličin po dobu minimálně 1 měsíce s nastavitelnou četností záznamu s minimální periodou 10s. Měřicí systém bude pro sledované vstupy zaznamenávat reálný čas.

Zaznamenávaná data (ukládána) na standardním paměťovém médiu budou mít tabulkový formát zpracovatelný na PC.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: tištěná

Seznam odborné literatury:

www stránky a katalogové listy výrobců číslicových obvodů

Brtník B., Číslicové systémy, BEN - technická literatura

Brtník B., Matoušek D., Algoritmy číslicového zpracování signálů, BEN -  
technická literatura

Hrabáček J., Komunikace mikrokontroléru s okolím díl 1 a 2, BEN - technická  
literatura

Vedoucí bakalářské práce:

**Ing. Tomáš Lelek, Ph.D.**


Katedra elektrotechniky, elektroniky a zabezpečovací  
techniky v dopravě

Datum zadání bakalářské práce: **4. ledna 2016**

Termín odevzdání bakalářské práce: **2. června 2017**

  
doc. Ing. Libor Švadlenka, Ph.D.  
děkan

L.S.

  
Ing. Dušan Čermák, Ph.D.  
vedoucí katedry

V Pardubicích dne 17. března 2017

## **Prohlášení**

Prohlašuji, že tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury. Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše. Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

Tato bakalářská práce byla realizována s využitím technologií Výukového a výzkumného centra v dopravě.

V Pardubicích dne 30.5.2017

Jakub Švadlenka

## **Poděkování**

Rád bych poděkoval, svým vedoucím práce, panu Doc. Ing. Ondřeji Černému, Ph.D. a panu Ing. Tomáši Lelkovi, Ph.D. za odbornou pomoc s tvorbou a cenné rady do realizace, panu Ing. Petru Sýkorovi za rady k realizaci. Poděkování patří i rodině, že mě po dobu studia podporovala.

## **Anotace**

Bakalářská práce se zabývá návrhem a realizací automatického systému pro záznam napětí, proudu a teploty pro 8 fotovoltaických panelů. Systém je určen pro statistické vyhodnocení energetické bilance. Popisuje návrh topologie sběrného systému v návaznosti na předchozí stav. V práci je obsažen popis návrhu elektrické, softwarové a mechanické části. Jsou zmíněny také možnosti budoucího rozšíření systému po softwarové stránce z hlediska zobrazení získaných údajů, případně základní výpočetní zpracování. Následně je popisována realizace, testování a ladění systému.

## **Klíčová slova**

záznam dat, embedded systém, datová sběrnice, galvanické oddělení, fotovoltaický panel

## **Annotation**

Bachelor thesis deals with design and implementation of automatic system for recording voltage, current and temperature for 8 photovoltaic panels. The system is designed for statistical evaluation of the energy balance. Describes the design of the collection system topology in relation to the previous state. The paper describes the design of electrical, software and mechanical parts. There are also mentioned the possibilities of future expansion of the system on the software side in terms of displaying the acquired data, or basic computational processing. The implementation, testing and debugging of the system is then described.

## **Keywords**

Data logging, embedded system, data bus, galvanic isolation, photovoltaic panel

# Obsah

ÚVOD.....	9
1. NÁVRH HARDWAROVÉ ČÁSTI.....	10
1.1 Výběr použitých obvodů.....	10
1.2 Návrh měřicí desky.....	13
1.2.1 Měřicí obvod.....	13
1.2.2 Předzesilovač a filtr.....	16
1.2.3 Analogově-digitální převodník.....	17
1.2.4 Galvanický oddělovač s optočleny.....	18
1.2.5 DC-DC izolovaný měnič.....	20
1.3 Návrh základní desky.....	20
2. Výpočty hodnot součástek.....	23
2.1 Výpočty vstupních obvodů.....	23
2.2 Výpočty galvanických opto-oddělovačů.....	24
2.3 DC – DC měnič.....	26
3. NÁVRH SOFTWAREOVÉ ČÁSTI.....	28
3.1 Popis operačního systému.....	28
3.1.1 Jádro a adresářová struktura.....	28
3.1.2 Souborové systémy.....	30
3.1.3 Služby.....	32
3.1.4 BASH.....	33
3.1.5 Síť a nastavení času.....	36
3.2 Popis ostatních komponent.....	36
3.2.1 Správa databáze RRDTool.....	36
3.2.2 Webový server Lighttpd.....	37
3.3 Sběrnice a komunikace s okolím.....	38

3.3.1 Sběrnice SPI a komunikace s A/D převodníky.....	38
3.3.2 Sběrnice 1-Wire a měření teploty.....	40
4. REALIZACE SOFTWAREVÉ ČÁSTI.....	40
4.1 Spuštění.....	41
4.2 Skript měření a záznamu dat.....	41
4.3 Získání hodnoty teploty.....	42
5. Naměřené údaje.....	43
6. ZÁVĚR.....	45
7. Seznam použité literatury.....	46



# ÚVOD

Tato bakalářská práce se zabývá návrhem a realizací měřicího a záznamového zařízení pro záznam dat z 8 fotovoltaických panelů. Konkrétně se zaznamenávají 3 fyzikální veličiny, napětí, proud a teplota. Fotovoltaické panely se v posledních desetiletích stále více prosazují ve výrobě elektrické energie. Zařízení bylo navrženo pro měření fotovoltaiky, nicméně jeho využití může být širší, prakticky se jedná o záznam šestnácti kanálů, tvořících dvojice, které jsou galvanicky oddělené od ostatních dvojic a záznam devíti teplot.

V areálu Výukového a výzkumného centra v dopravě v Doubravících je nainstalováno celkem 8 fotovoltaických panelů určených k výuce. Tento systém měří a zaznamenává provozní hodnoty do souboru na SD kartu a z hodnot následně vytváří textový i grafický výstup vhodný pro prezentaci, kde textový výstup je statistického charakteru a může popisovat například celkové množství vyrobené energie za konkrétní časové období, například statistika posledního měsíce. Tuto statistiku pak doplňují grafy, kde jsou údaje vidět v časovém kontextu a je možné srovnávat.

K realizaci jsem zvolil mikropočítač Raspberry Pi. Tento mikropočítač je ve své polední verzi která byla pro účely tohoto zařízení pořízena, osazen čtyř jádrovým 64-bitovým procesorem Broadcom BCM2837 o taktu 1.2GHz s 1GB operační paměti RAM.

Zvolením tohoto mikropočítače se zpřístupnily i další možnosti v prezentaci naměřených dat. Díky vyššímu výkonu je možné provozovat na tomto mikropočítači webový server a publikovat tak i data ve formě vhodné k prezentaci i široké veřejnosti. Například při dnech otevřených dveří, nebo podobných akcích, je možné zobrazit naměřené údaje prostřednictvím počítačové sítě i jinde, než přímo v místě instalace měřicího zařízení. K síti se nejnovější generace Raspberry Pi ve verzi 3 B+, může kromě dosavadní možnosti připojení k síti LAN přes kabel s konektorem RJ-45, také připojit prostřednictvím technologie bezdrátových sítí Wi-Fi.. Doplněním těchto možností je pak spuštění samotného webového prohlížeče i přímo v mikropočítači a prezentací dat přímo přes zobrazovač, například televizor či monitor připojený přes rozhraní HDMI.

# 1. NÁVRH HARDWAROVÉ ČÁSTI

## 1.1 Výběr použitých obvodů

Při návrhu hardwarové části jsem vycházel z požadavků na zadané vlastnosti. Aby nebylo nutné pro zařízení konstruovat další kryt rozhodl jsem se pro koncepci, kterou lze zabudovat do stávající rozvodné skříňky se svorkovnicemi od jednotlivých panelů, která byla již nainstalována dříve.

Zařízení je koncipováno jako měřicí moduly, které se zasouvají do konektorů na základní desce, ke které je připojen řídicí mikropočítač. Propojení mezi základní deskou a mikropočítačem je realizováno plochým mnohažilovým kabelem. Celý přístroj je pak možné například zabudovat do již výše zmíněné rozvodné skříňky pomocí distančních sloupků. Případně může být přístroj vestavěn do vlastního krytu a později rozšířen, nebo využit po patřičné úpravě i pro jiný účel než je měření fotovoltaických panelů.

Během návrhu přicházely v úvahu různá řešení zadání této práce a to totiž návrh zařízení, které bude schopno zaznamenávat 8 vstupních kanálů pro měření proudu, 8 kanálů pro měření napětí a 9 hodnot teploty. Měření teploty už od počátku bylo uvažováno přenechat teplotním čidlům DS18B20. Toto čidlo se připojuje k digitální sběrnici 1-Wire, která umožňuje připojit 9 těchto čidel k jedné datové sběrnici. K měření napětí a proudu je zapotřebí obě hodnoty převést na napětí nižší, než je maximální dovolené napětí na vstupu následujících obvodů předzesilovače. Pro měření jsem na vstup zařadil nastavitelný napěťový dělič a pro měření proudu snímací bočník, ze kterého je odebíráno napětí odpovídající proudu procházejícímu měřicími svorkami.

Dále bylo potřeba vyřešit požadavek na galvanické oddělení, přičemž byly zvažovány různé možnosti. Bylo třeba rozhodnout se jestli dojde ke galvanickému oddělení na straně analogové, nebo na straně digitální. Pro galvanické oddělení nasnímané analogové hodnoty je zapotřebí izolačního zesilovače, nebo optočlenu. Varianta s izolačním zesilovačem ve formě zakoupené součástky připadala v úvahu, neboť u nich výrobce garantuje přesnost převodu měřené hodnoty. Jsou na trhu rovněž izolační zesilovače, které v sobě spojují funkci programovatelného předzesilovače, analogově-digitálního převodníku a izolovaného

napájecího zdroje, čímž by se značně zjednodušil návrh a zmenšila velikost desky s plošnými spoji. Hlavní nevýhodou varianty s izolačními zesilovači oproti variantě popsané dále, je vysoká cena za izolační zesilovač. Další varianta je sestavit obvod s optočleny jejichž převodní charakteristika bude pokud možno co nejvíc lineární. I přes využití součástek které jsou charakterizovány téměř lineárním průběhem, je potřeba zapojit dva tyto optočleny, ve vhodném zapojení. Galvanické oddělení v analogové části jsem pro další návrh neuvažoval pro jeho vyšší náročnost jak na návrh, tak na použité součástky. Rozhodl jsem se realizovat oddělení na straně digitální. Tím odpadá potřeba nějak více zkoumat a řešit průběh převodní charakteristiky optočlenu, neboť je přenášena informace vyjádřena pouze dvěma stavy, logická 1 a logická 0.

Oddělení na straně digitální sebou nese potřebu zvolit obvodové prvky, optočleny a tranzistory tak, aby zvládaly přenášet hrany digitálních impulzů s dostatečnou strmostí a dostačující opakovací frekvencí. Pro oddělení digitální sběrnice je tedy vhodné použít optočleny rychlé. Další možností je využít běžné optočleny a snížit hodinový kmitočet digitální komunikační sběrnice. Dojde tím samozřejmě ke zpomalení komunikace, vzhledem k požadavku na nejkratší interval 10 vteřin mezi jednotlivými záznamy je však i rychlost nejnižší doporučená pro daný analogově-digitální převodník, stále postačující pro splnění zadání.

Galvanicky oddělené obvody ale potřebují také napájecí zdroj, pro každou dvojici měřících vstupů, kde tyto dva vstupy již od sebe pro měření napětí a proudu. Výstup tohoto zdroje musí být galvanicky oddělen od napájení ostatních obvodů. Pro řešení této části obvodu jsem zpočátku prověřoval možnosti, které trh nabízí ve formě hotových izolovaných DC-DC měničů. Tyto měniče se vyskytují v pouzdrech, která se zaletují do plošného spoje stejně, jako by šlo o součástku. Rozhodl jsem se však pro stavbu vlastního izolovaného DC-DC měniče z diskretních součástek a integrovaného obvodu MC34063. Důvod je opět ten, že tato varianta vychází při počtu osmi kusů výrazně levněji.

Po konzultaci ve VVCD v Doubravicích jsem zjistil že panely se nebudou zapojovat všechny do série, neboť nejsou stejného typu, tudíž není požadavek na zvýšenou izolační pevnost. V případě, že by došlo v budoucnu k požadavku na sériové spojení všech 8 panelů do série, může nejvyšší napěťový potenciál v zařízení být až okolo 300V a zařízení tak, jak je, pro toto izolační napětí navrženo není.

Naměřenou hodnotu, která je v podobě analogového signálu, je třeba pro další účely převést do digitální podoby. K tomu slouží analogově-digitální převodník. Byly zvažovány celkem tři varianty. Varianta první spočívala v nápadu využít jeden, případně dva analogově-digitální převodníky a analogový multiplexer, které by přepínaly galvanicky oddělený a přizpůsobený analogový signál jdoucí ze vstupních obvodů, přičemž se počítalo, že galvanické oddělení bude na analogové straně.

Druhá uvažovaná varianta spočívala v náhradě multiplexeru a jedno či dvoukanálového A/D převodníku dvěma integrovanými obvody MCP3208. Tento integrovaný obvod v sobě sdružuje dvanáctibitový analogově-digitální převodník a osmikanálový multiplexer. Komunikace probíhá po sběrnici SPI, která je typu Master-Slave, což znamená, že je jedno zařízení hlavní (Master) a ostatní zařízení jsou v komunikaci na podružné pozici. S tímto druhem sběrnice souvisí nutnost možnosti zvolit Slave zařízení, se kterým chce Master zařízení komunikovat, to by se v případě dvou zařízení na sběrnici SPI provedlo vyvedením dvou samostatných pinů z řídicího mikropočítače pro funkci výběru zařízení (Chip select), případně vyvedením jednoho pinu, který je potřebný pro komunikaci i s jedním tímto obvodem, následnou logickou negací ať už tranzistorem v zapojení se společným emitorem, nebo invertorem v provedení jako hradlo integrovaného obvodu, dostaneme signál použitelný pro výběr druhého obvodu.

Třetí varianta, kterou jsem nakonec vybral pro realizaci, spočívá ve využití osmi kusů dvoukanálové verze obvodu zmíněného výše. Tento obvod má označení MCP3202 a obsahuje ve svém pouzdru dvoukanálový multiplexer a dvanáctibitový analogově-digitální převodník a stejně jako typ MCP3208 i typ MCP3202 komunikuje po sběrnici SPI, tudíž je opět potřebné řešit jak se mezi osmi obvody bude přepínat zrovna ten, se kterým se bude komunikovat.

Jako první jsem prozkoumával variantu bez adresního řadiče, tak zvaný režim Daisy-chain, což je vlastně zřetěžením všech SPI zařízení za sebou. Řídicí mikropočítač je výstupem dat SPI připojen na vstup prvního obvodu. Výstupní data prvního obvodu jsou připojena na vstup druhého, z něj výstup opět na vstup třetího a tak dále. A z posledního obvodu se data zavedou do mikropočítače. Vznikne tak sběrnice sériově zapojená, která se chová jako posuvný registr. Toto řešení jsem po bližším prozkoumání zavrhl, jelikož tato funkcionálna nefunguje napříč širokým spektrem různých typů zařízení se sběrnici SPI zcela spolehlivě a bylo přistoupeno

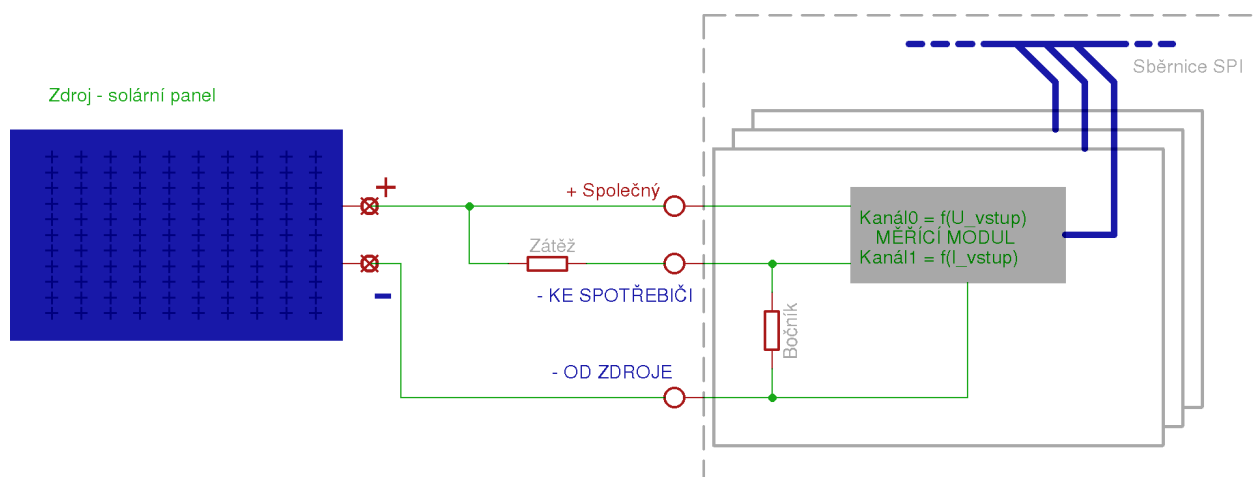
k prozkoumání možnosti sběrnici zapojit standartě, paralelně a osazení adresním voličem. Zde jsem vybíral mezi typy integrovaných obvodů 74HC595, což je posuvný registr, adresním dekodérem 74HC238 nebo obvodem 4017, což je pětistupňový Johnsonův čítač s hodinovým, blokovacím a nulovacím vstupem a výstupem s dekodérem 1 z 10. Volba nakonec padla na poslední jmenovaný integrovaný obvod typu CMOS řady 4000. Při spojení jen jednoho obvodu analogově-digitálního převodníku MCP3202 s piny pro SPI sběrnici řídicího mikropočítače Raspberry Pi, je využít směrem z Raspberry k převodníku jeden vodič výběru obvodu. Pro zapojení více převodníků s obvodem 4017 stačí piny dva, jeden pro volbu, který se zavede jako hodinový kmitočet do čítače 4017 a pin reset, kterým čítač vynulujeme. U obvodu 74HC238 je pro ovládání výstupů zapotřebí vodičů tří. U obvodu 74HC595 by pro mé účely stačily piny dva. Aplikování obvodu 4017 vycházelo nejjednodušeji, neboť kromě nutnosti doprogramovat obsluhu nulování nevyžadoval jiný zásah do řídicího programu oproti komunikaci s jedním obvodem MCP3202. Při vyčítání jednoho obvodu přechází pin sběrnice SPI pro výběr obvodu (Chip-select) na okamžik čtení z vysoké úrovně do nízké, po ukončení komunikace se vrací zpět. Zapojením obvodu 4017 a více převodníků dochází i bez programové změny vůči opakovanému čtení jednoho převodníku, ke střídavému přepínání mezi obvody. Bez dalšího opatření není kontrola nad aktuálním výběrem a k tomu slouží pin reset – nulování, který uvede obvod do předem známého stavu.

Zkoumal jsem také řešení desek s plošnými spoji, jelikož by jedna deska vycházela s nevhodnými rozměry, bylo zvoleno řešení se základovou deskou, do které se zasouvá osm měřících desek přes dvouřadé konektory.

## **1.2 Návrh měřící desky**

Při návrhu jsem nejen měřící desky ale vůbec celého zařízení jsem vycházel obvykle z doporučených zapojení z typových listů příslušných integrovaných obvodů. Obvodové zapojení měřící desky se skládá z částí které zde jednotlivě popíši: měřící obvod, předzesilovač s filtrem, analogově-digitální převodník, DC-DC izolovaný měnič, galvanický oddělovač s optočleny.

## 1.2.1 Měřicí obvod

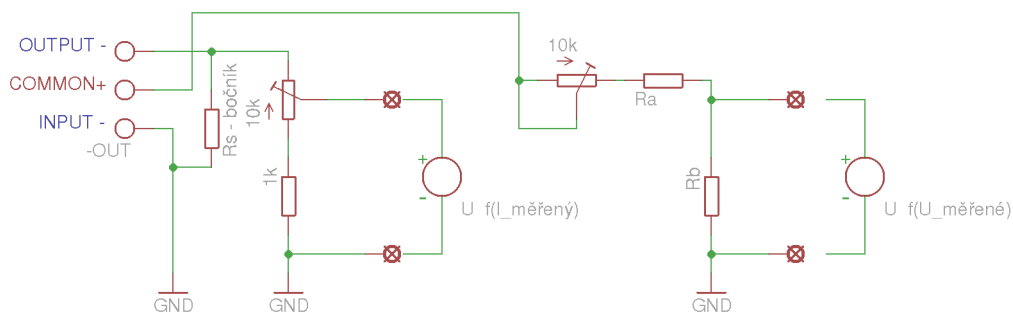


Obr. 1 Schéma zapojení zařízení do měřeného obvodu

Navrhované zařízení má měřit napětí, proud a teplotu. K měření napětí a proudu v rámci požadovaných parametrů, je zapotřebí vybavit každou kartu hned u měřících svorek měřícím obvodem, který převede napětí solárního panelu s nějakou rezervou na napětí nižší. Napětí za tímto vstupním obvodem by mělo být sníženo nejvýše na úroveň napájecího napětí za DC-DC měničem, což je maximální rozsah napětí pro zpracování následným obvodem předzesilovače. Jak je uvedeno v katalogovém listu obvodu předzesilovače MCP602, maximální rozpětí vstupních napětí na analogových vstupech je o 1 V větší, než napájecí napětí a to kladná i záporná větev. Náš obvod pracuje s nesymetrickým napájením, tudíž zápornou větví je v tomto případě potenciál 0V (svorka minus pólu vstupu). Skutečný použitelný rozsah napětí která jsou na vstupu předzesilovače je navíc dále omezen zesílením předzesilovače.

Měřené napětí se bude obvykle pohybovat do 40 V, nicméně jsou hodnoty součástek navrženy kompromisně tak, aby byl přístroj přizpůsobitelný pro měření i jiného rozsahu. Je vhodné myslet i na nějakou rezervu v napěťovém rozsahu, ale zároveň je snaha využít bitovou hloubku zpracování použitého analogově-digitálního převodníku. Jako maximální měřitelnou hodnotu jsem určil 80 V. Snížení vstupního napětí na napětí vhodné k dalšímu zpracování se realizuje vhodným napěťovým děličem. Tento dělič by měl být nastavitelný, proto je vybaven potenciometrickým trimrem. Míra nastavitelnosti vstupního rozsahu je na druhou stranu omezena požadavkem na přesnost a mechanickou stálost tohoto nastavení. Je tedy vhodné při výběru hodnot součástek mít na paměti, že nastavení napěťového děliče

na vstupu se děje potenciometrickým trimrem a požadavek na nastavitelnost vstupního rozsahu je tak omezen spíše na kalibraci v menším rozsahu. Pro větší změnu dělicího poměru vstupních obvodů doporučuji změnu hodnot rezistorů s drátovými vývody na vstupním děliči a ve zpětné vazbě. Při vložení větší ohmické hodnoty trimru pro požadavek větší nastavitelnosti bez úpravy zapojení by docházelo například vlivem byť malých otřesů k samovolným změnám nastavení, řešením by bylo použít mnoho-otáčkový trimr, který je ale příliš drahý. Bude tedy vhodné volit menší hodnotu trimru a v případě že by přesto docházelo k samovolným změnám nastavení, které by neprojevovaly závislost na teplotě, doporučuji u nastalo nastaveného zařízení zvážít zakápnutí nastavitelné části trimru k tělu lakem, či lepidlem.



Obr. 2 Schéma zapojení vstupních obvodů

Vysvětlivka :Common – společný ( + pól )

Input – proudová svorka od zdroje ( - pól ), Output – proudová svorka ke spotřebiči ( - pól )

Napětí na výstupu  $U_{RB}$  nezátíženého napěťového děliče z odporů  $R_A$  a  $R_B$  lze vypočítat:

$$U_{RB} = U_{VSTUPNI} \cdot \frac{R_B}{R_A + R_B} \quad (1.1)$$

K měření proudu lze použít proudový snímač využívající Hallův článek, nebo převod na napětí na bočníku. Měření proudu Hallovým snímačem jsem nezvolil z důvodu vysoké složitosti a ceny.

Měření proudu lze provést mnohem snadněji a to tak, že jej necháme procházet bočníkem, což je výkonový rezistor typicky malé hodnoty. Měření proudu bočníkem má výhodu v tom, že je to řešení levné a tradičně známé. Nevýhodu má v tom, že procházející proud způsobí

úbytek výstupního napětí podle vztahu:

$$U_s = R_s \cdot I_m \quad (1.2)$$

S tímto úbytkem  $U_s$ , který je nutný, jelikož je to dále zpracovávaná veličina, která lineárně odpovídá procházejícímu proudu  $I_m$ , souvisí pokles svorkového napětí za měřicím zařízením o tento napěťový úbytek a také ztrátový výkon  $P_Z$  na bočniku, který se odvede ve formě tepla. Ztrátový výkon  $P_Z$  lze vypočítat vztahem:

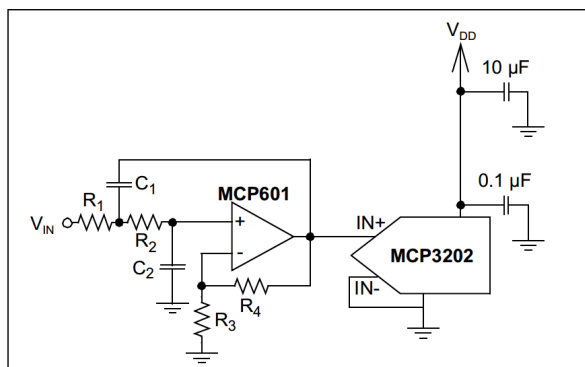
$$P_Z = I^2 \cdot R_s \quad (1.3)$$

O tento ztrátový výkon je snížena užitečná energie, je tedy snaha navrhnout bočník tak, aby v poměru k přenášenému výkonu byly ztráty poměrně malé. Účinky ztrátového výkonu budou v ohřívání prostoru celého měřicího zařízení, u něhož navíc předpokládám trvalý provoz. Z tohoto důvodu je vhodnější bočník s nižším odporem. Naopak z hlediska dalšího zpracování je vhodné aby úbytek napětí na bočniku nebyl až příliš malý, neboť potom bude nutné u předzesilovače nastavit vyšší zesílení a to může vést k nestabilitám v podobě samovolného kmitání aktivního filtru, který je součástí předzesilovače, tak k ovlivňování absolutní chyby okolní teplotou, která je ovlivňována ztrátovým výkonem na bočniku. Výběr hodnoty bočniku, tím pádem i zesílení předzesilovače s aktivním filtrem a hodnoty konstant pro přepočítání v softwaru zařízení, bude věcí kompromisu.

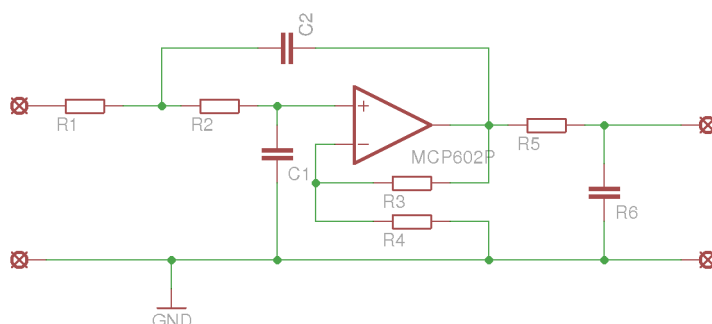
### 1.2.2 Předzesilovač a filtr

Pro předzesilovač byl vybrán dvojitý operační zesilovač MCP602. Tento operační zesilovač výrobce přímo zmiňuje i v katalogovém listu použitého analogově-digitálního převodníku. Tento obvod se vyznačuje tím, že má vlastnost rail-to-rail, to znamená, že vstupní i výstupní napětí může nabývat až hodnot napětí napájecích větví, což u běžného operačního zesilovače bez této vlastnosti neplatí. Vycházel jsem z aplikační poznámky z katalogového listu obvodu MCP3202.





Obr. 3 Zapojení analogové části doporučené výrobcem



Obr. 4 Schéma obecného zapojení předzesilovače s aktivním filtrem

Prvním požadavkem na vstupní obvod je signál vyfiltrovat od kmitočtů vyšších, než je maximální uvažovaný kmitočet vstupního signálu, který chceme ještě zaznamenat, pro tento proces se používá název anti-aliasing. Druhým požadavkem je přinejmenším u kanálu měřicího proud, signál zesílit. Filtr je aktivní, druhého řádu, typu dolní propust. Za předpokladu, že  $R_1 = R_2$  a  $C_1 = C_2$ , se mezní kmitočet tohoto filtru určí ze vztahu:

$$f = \frac{1}{2\pi RC} \quad (1.4)$$

Ve zpětné vazbě je obvod osazen dvojicí rezistorů, která určuje jeho zesílení. Nastavení zesílení bude moci probíhat jen v rozsahu takovém, aby nedošlo k samovolnému rozkmitání.

Pro tento typ filtru se uvádí hranice rozkmitání  $A = \sim 3$

Zesílení určíme ze vztahu:

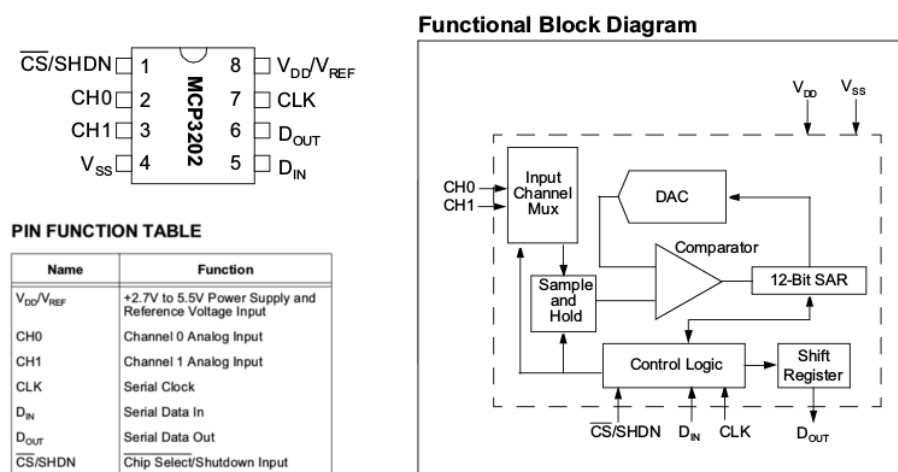
$$A = 1 + \frac{R_3}{R_4} \quad (1.5)$$

Pro rozšíření možnosti úpravy rozsahu je několik rezistorů technologie SMT nahrazeno standardními vývodovými rezistory THT, které lze snadněji vyměnit při případné změně účelu

zařízení. V případě že by pro měření proudu nedostačovalo zesílení předzesilovače kombinovaného s aktivní dolní propustí, je možné přistoupit i k takové úpravě, že se zapojí obě poloviny obvodu za sebe, přičemž se u jednoho, který bude fungovat jako předzesilovač se neosadí součástky upravující frekvenční charakteristiku tj. oba kondenzátory a pak je možné neosadit a přemostit i oba rezistory ve vstupu operačního zesilovače a zůstanou jen rezistory ve zpětné vazbě. Druhá polovina obvodu MCP602 se zapojí za první s tou úpravou, že se vypustí odpory ve zpětné vazbě a na místo rezistoru z výstupu operačního zesilovače do invertujícího vstupu se zapojí drátová propojka, čímž vznikne dolní propust s jednotkovým zesílením. Kanál měřící napětí se v tomto případě spojí přímo ze vstupních obvodů na filtrační RC člen na vstupu analogově-digitálního převodníku. Jiná, mnohem rychlejší varianta úpravy obvodu je vynechání kondenzátoru ve zpětné vazbě, čímž se z obvodu stane pasivní dolní propust 1. řádu s připojeným zesilovačem.

### 1.2.3 Analogově-digitální převodník

Naměřená hodnota, vyfiltrovaná a upravená na patřičnou úroveň, se podstupuje dalšímu zpracování v analogově-digitálním převodníku. Pro účely mé práce mi byl doporučen vedoucím práce obvod MCP3202. Integrovaný obvod MCP3202 je dvanáctibitový analogově-digitální převodník. Převod probíhá metodou postupné aproximace. Obvod pracuje v poměrně širokém rozmezí napájecího napětí 2,7 – 5,5 V, což umožňuje jeho bezpečné použití jak s 3,3V tak s 5V logikou. Odběr obvodu je nízký, při napájení 5V, při měření odebírá 0,5 mA a 5 $\mu$ A je maximální odběr v neaktivním stavu.



Obr. 5 Analogově-digitální převodník MCP3202, rozložení a funkční význam pinů.

Vnitřní bloková struktura obvodu.

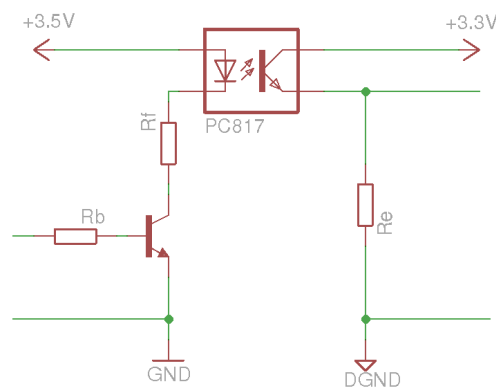
Vstupní kanály jsou dva, ale nejedná se o dva samostatné analogově-digitální převodníky, ale o převodník jeden a na vstupu obvodu je multiplexer, který je přepínán řídicí logikou obvodu, na základě instrukcí přijatých po sběrnici SPI. Tyto dva vstupy mohou pracovat jako dva nezávislé kanály, nebo i jako jeden pseudo-diferenciální vstupní vstup, kde pak dostaneme hodnotu napětí mezi těmito dvěma vstupy. Převedená hodnota  $N_{ADC}$  odpovídá napětí na vstupu analogově-digitálního převodníku  $U_{ADC}$  dle vztahu:

$$N_{ADC} = \frac{4096 \cdot U_{ADC}}{U_{Napájecí}} \quad (1.6)$$

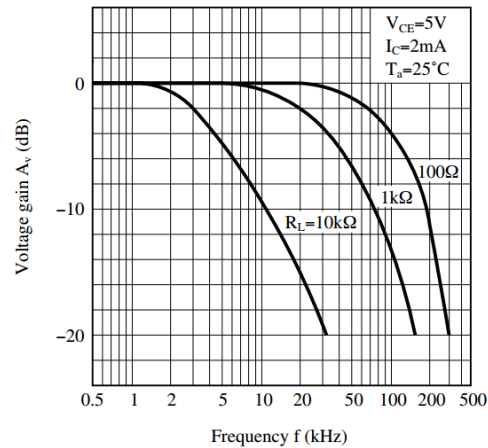
Převedená hodnota je poté přes galvanický oddělovač přenesena na sběrnici SPI řídicího mikropočítače, kde je dále zpracovávána.

#### 1.2.4 Galvanický oddělovač s optočleny

Tím jak se postupně formovaly a upřesňovaly požadavky na navrhované zařízení, bylo čím dál víc patrné, že zařízení bude muset mít každou dvojici kanálů galvanicky oddělenou. Galvanické oddělení bude muset poskytovat jak přenos energie směrem k měřicí části, tak i obousměrný přenos dat, pro komunikaci analogově-digitálního převodníku s řídicím mikropočítačem po sběrnici. Pro přenos dat s galvanickým oddělením lze využít indukční cestu, s oddělovacím transformátorem, což by ale vyžadovalo další obvody, nebo optickou cestu. Já vybral oddělení optické, jakožto obvodově nejsnadnější. Pro moji práci jsem vybral optočlen typu PC817.



Obr. 6 Základní zapojení jednoho optického oddělovače



Obr. 7 Graf závislosti zesílení na přenášené frekvenci

Z grafu závislosti zesílení na přenášené frekvenci je patrné, že běžné hodinové kmitočty, které se používají pro sběrnici SPI, které jsou v základním nastavení obvykle na hodnotách 100 nebo 500 kHz, jsou příliš vysoké pro přenos tímto typem optočlenu. Analogově-digitální převodník MCP3202 umožňuje podle katalogového listu přenos po sběrnici s nejnižším hodinovým kmitočtem 10kHz. Nižší frekvenci už výrobce nedoporučuje, mohlo by u tohoto převodníku zvláště při zvýšené teplotě docházet k nelineárním chybám v měření, protože kondenzátor ve vzorkovací části převodníku by neudržel tak dlouhou dobu náboj ve stejné výši. Na dodržení zadání práce, z hlediska nejkratšího intervalu mezi měřením i tato nejnižší rychlost zcela postačuje.

Způsob zapojení rezistoru v tranzistorové části optočlenu, s rezistorem v emitoru, nebo s rezistorem v kolektoru, má vliv na polaritu přenášeného signálu. Pokud bude rezistor připojen na kladný pól, druhým vývodem spojen s kolektorem a emitor připojen na záporný pól, bude signál jdoucí ze společného uzlu rezistoru a optočlenu invertován. Toho je využito pro signál výběru. Pro neinvertovaný signál budou tyto dva obvodové elementy prohozeny s tím, že kolektor je spojen s kladným pólem, z emitoru jde výstup a je spojen s rezistorem, který pokračuje na záporný pól. Toto zapojení je použito u všech ostatních signálů. Z hlediska dynamiky přenášeného signálu jsou obě zapojení rovnocenná.

Velikost rezistoru  $R_e$  (Obr.6) a  $R_L$  (Obr.7), který je zapojen jako zátěž optočlenu, je volena s ohledem na napájecí napětí, odebíraný proud a potřebnou dynamiku přenosu signálu. Při příliš velké hodnotě tohoto odporu začne být přenos nespolehlivý v důsledku deformací signálu.

### 1.2.5 DC-DC izolovaný měnič

Je zapotřebí galvanicky oddělenou část také napájet. Toto napájení je provedeno DC-DC měničem. Bylo přistoupeno ke stavbě měniče s integrovaným obvodem MC34063A. Tento obvod je určen pro zvyšující, snižující a invertující měniče. Pro měnič izolační jsem nenašel aplikační poznámku, nicméně z principu obvodu není důvod, aby měnič této topologie s tímto obvodem nefungoval. Měnič je blokující, se zpětnou vazbou optočlenem. Pro galvanicky oddělený přenos energie mám v plánu využít feritová odrušovací jádra, jelikož jde o dostupný materiál, který je v této aplikaci použitelný, protože jde o malý přenášený výkon.

### 1.3 Návrh základní desky

Základní deska slouží k propojení měřících desek s řídicím mikropočítačem, k adresaci těchto desek a je na ní vyveden konektor pro připojení sběrnice 1-Wire s teplotními čidly.

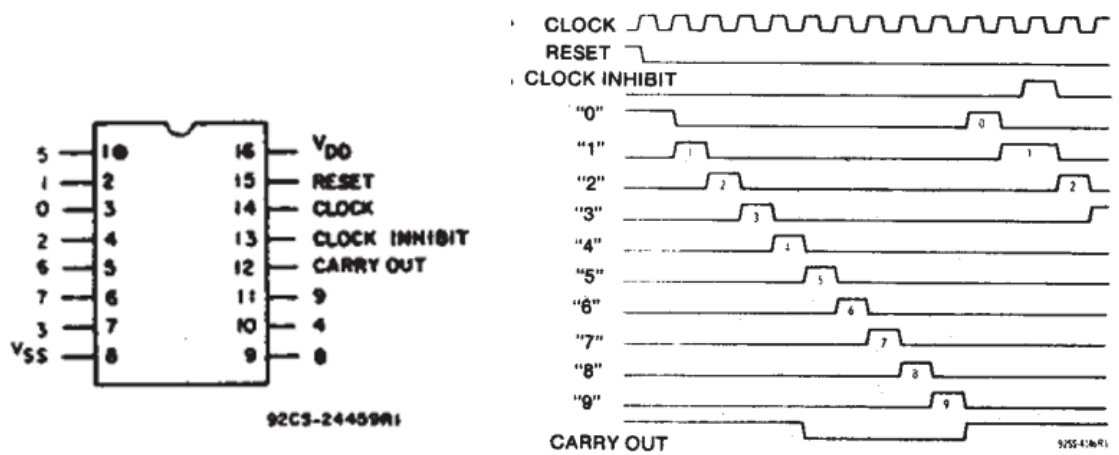
Pro připojení měřících karet je základní deska osazena dvouřadými zásuvkami v rastru 2,54mm o celkovém počtu dvanácti pinů na zásuvku. Do těchto zásuvek jsou zasunuty měřící desky. Dvouřadé uspořádání bylo zvoleno pro mechanické zesílení a zvýšení spolehlivosti a životnosti. Konektory nicméně nemohou nést síly, které vzniknou při běžném používání zařízení, samy o sobě, proto jsou měřící desky ještě spojeny skrz otvor v horní straně desky závitovou tyčí a matkami o průměru 3mm. Taková sestava je při zabudování do krytu přístroje a spojení desek konstrukčními prvky do stěn krytu už pravděpodobně dostatečně mechanicky pevná.

Konektor s vyvedeným vodičem sítě 1-Wire pro připojení teplotních čidel obsahuje také napájecí piny, pro zamezení připojení v nesprávné polaritě je na 3 piny použit čtyř-pinový konektor s tím, že je jeden z prostředních pinů v konektoru vyjmut, vynechán a v opačném konektoru je vynechaný pin zaslepen. Stejné opatření je i u GPIO na straně základní desky. Polaritu 40-pinového konektoru GPIO na Raspberry Pi 3 je potřeba hlídat při manipulaci s tímto spojem.

Dále pak deska obsahuje dva tranzistory a dva rezistory, pro proudové posílení GPIO výstupu, neboť bude muset budít všech 8 optočlenů, při všech zapojených měřících deskách. Tyto dva tranzistory patří k oddělovačům zmíněným v části 1.1.4 Galvanický oddělovač s optočleny.

Na desce je v patici osazen integrovaný obvod typu CMOS CD4017B. Tento obvod řeší výběr jednotlivých analogově-digitálních převodníků na měřicích deskách, pomocí signálu /CS což je zkratka pro invertovaný „výběr čipu“. Na straně analogově-digitálního převodníku je signál běžně ošetřen pull-up rezistorem, tj. rezistor zapojený jako kolektorový vůči N nebo NPN tranzistoru. Sepnutím tohoto tranzistoru napětí na pinu /CS poklesne a obvod analogově-digitálního převodníku se aktivuje.

Jednotlivé signály výběru analogově-digitálního převodníku jsou vedeny z výstupů obvodu 4017. Výstup 0 není využit, neboť jde o pin, který je aktivní po resetu.



Obr. 8 Integrovaný obvod typu 4017 – rozložení vývodů a časový diagram funkce obvodu

Integrovaný obvod CD4017BE, je logickým obvodem technologie CMOS z řady 4000. Jedná se o pětinasobný Johnsonův čítač s dekadickým výstupem 1 z 10. Obvod je doporučeno napájet od 3 V do 18 V a spotřeba samotného obvodu je v klidovém stavu a při napájení 5 V typicky 0,04  $\mu\text{A}$ , maximálně však 5  $\mu\text{A}$ . Ovládání obvodu probíhá třemi logickými vstupy. Vstup nulování, pin RESET (15) slouží k vynulování čítače, nulování probíhá přivedením vysoké úrovně na tento vstup. Vstup CLOCK (14) je hodinový, udává takt obvodu a reaguje na vzestupnou hranu hodinového signálu. V integrovaném obvodu je za tímto vstupem Schmittovo hradlo, které zajišťuje potřebnou hysterezi. Vstup CLOCK INHIBIT (13) má za účel znečitlivit obvod na hodinový vstup, pin CLOCK (14). K přičtení čítače a tudíž k přepnutí z jednoho výstupu na další v pořadí dojde za podmínky, že pin CLOCK INHIBIT (13) se nachází v nízké úrovni a pin CLOCK (14) zrovna přešel z nízké do vysoké úrovně.

Z výstupů tohoto obvodu se ovládají optočleny galvanického oddělovače. Signál výběru je

potřeba ještě logicky invertovat. To je zajištěno zapojením tranzistorové části optočlenu s emitorem na záporném pólu a kolektorovým rezistorem připojeným na kladnou napájecí větev.

Základní deska též má proudově posílit výstupy z řídicího mikropočítače, konkrétně se jedná o hodinový signál a výstup dat z řídicího mikropočítače. Pro tento účel je na základní desce zapojen dvakrát obvod zesilovače se společným emitorem, v kolektoru má zapojeny příslušné optočlenu připojených měřících karet. Jedná se o část obvodu galvanického oddělovače (Obr. 6)

Připojení základní desky k řídicímu počítači je provedeno plochým kabelem o 26 vodičích. Starší generace Raspberry Pi má GPIO konektor právě o 26 pinech, tudíž se jedná o formu kompatibility. Samořezné konektory na tyto ploché kabely mají podélné okraje příliš široké na to, aby šel bez úprav 26 pinový konektor zasunout do konektoru 40 pinového a ten se zase nevejde do Raspberry starší generace, neboť v tom překáží další konektory na Raspberry. Je tedy zapotřebí zvolit správný konektor ke správné generaci Raspberry.

Z řídicího mikropočítače Raspberry Pi je celý přístroj skrze GPIO konektor napájen. Raspberry Pi poskytuje 5V, kterými je samo napájeno a 3,3V přes vnitřní regulátor. Vstupně výstupní piny jsou v úrovních 3,3V a nejsou tolerantní vůči 5V. Proud odebíraný z 3,3V větve k napájení vlastní aplikace skrze GPIO konektor je u starší generace, která využívala lineární stabilizátor doporučeno držet do 50mA. Nová generace Raspberry Pi má stabilizátor spínaný a je možno odebírat až 500mA dle schopností napájecího zdroje.



Obr. 9 Raspberry Pi 3 B+.

## 2. Výpočty hodnot součástí

K realizaci ze skutečných součástí je z hlediska hodnot rezistorů a kondenzátorů třeba přistupovat poněkud jinak. Nejen z hlediska výroby je vhodné směřovat výběr hodnot k jejich unifikaci. V této části vypočítám provozní vlastnosti spojení jednotlivých funkčních bloků. Budu v popisu opět postupovat po směru měřeného signálu.

### 2.1 Výpočty vstupních obvodů

Zde uvádím výpočet stejnosměrných parametrů obvodu, výpočet je proveden pro uvažované maximální napětí 80 V a maximální proud 12 A .

Napětí na výstupu napěťového děliče

-s trimrem staženým do nulové hodnoty

$$U_{RBmin} = U_{VSTUPNÍ} \cdot \frac{R_B}{R_A + R_B} = 80 \frac{1 \cdot 10^3}{20 \cdot 10^3 + 1 \cdot 10^3} = 3,809 \text{ V} \quad (2.1)$$

- maximální nastavení trimru (zařazena celá jeho hodnota)

$$U_{RBmax} = U_{VSTUPNÍ} \cdot \frac{R_B}{R_A + R_{TRIMR} + R_B} = 50 \frac{1 \cdot 10^3}{20 \cdot 10^3 + 10 \cdot 10^3 + 1 \cdot 10^3} = 2,58 \text{ V} \quad (2.2)$$

Následný obvod předzesilovače a aktivního filtru byl nastaven do jednotkového zesílení, nahrazením rezistoru  $R_3$  drátovou propojkou a vynecháním rezistoru  $R_4$ . Napětí na výstupu napěťového děliče by tedy mělo být stejné jako na vstupu analogově digitálního převodníku. Ten má 4096 kroků a měřené napětí porovnává se svým napájecím napětím, které tím pádem slouží jako referenční, o hodnotě 3,5V. Zvolil jsem softwarové násobení hodnotou 20 a dostal jsem rozsah vstupního měřeného napětí 81 900mV.

Napětí na snímacím bočníku proudu

Původní varianta bočníku, počítala s využitím kanthalového odporového drátu, ovšem



přišel jsem na to, že odpor  $4,75 \Omega / m$  je příliš velký. Další překážkou je to, že kanthal se dá jen nesnadno pocínovat, uvažovalo se o variantě, kdy by se cínované konce krátce ponořily před samotnou operací do kyseliny fosforečné.

Po prozkoumání dostupných materiálů jsem došel k použití měděného drátu o průřezu  $0,75\text{mm}^2$ . Konkrétně se jedná o drát z kabelu rozvodů počítačových sítí UTP  $4 \times 2 \times 0,75\text{mm}^2$ , s měrným odporem přibližně  $0,634 \Omega/m$ . Pro bočník jsem zvolil hodnotu  $25\text{m} \Omega$ , které odpovídají asi 4cm drátu.

Instalované solární panely mají psaný nejvyšší provozní proud 8A. Pro rozsah analogově-digitálního převodníku jsem zvolil hodnotu 1,5 násobně vyšší – 12A, která by však rozhodně neměla být trvale využívána, bočník je dimenzován maximálně na 8A.

$$U_s = R_s \cdot I_{max} = 0,025 \cdot 12 = 0,3V \quad (2.2)$$

Obslužný software násobí hodnotu z analogově digitálního převodníku u kanálu měřícího proud hodnotou 3, rozsah bude tedy do 12 285mA. Napětí z bočníku je potřeba napěťově zesílit pro využití bitového rozsahu analogově-digitálního převodníku.

Zesílení operačního zesilovače určíme ze vztahu:

$$A = 1 + \frac{R_3}{R_4} = 1 + \frac{10 \cdot 10^3}{1 \cdot 10^3} = 11 \quad (2.3)$$

Mezní frekvence aktivního filtru

$$f = \frac{1}{2\pi RC} = \frac{1}{2\pi \cdot 47 \cdot 10^3 \cdot 100 \cdot 10^{-9}} = 34 \text{ Hz} \quad (2.4)$$

## 2.2 Výpočty galvanických opto-oddělovačů

Hodnoty rezistorů každého jednotlivého galvanického oddělovače jsou navrženy s ohledem na konkrétní obvodový celek. Záleží, jakým napětím je napájena část s tranzistorem, rezistorem  $R_b$  a  $R_f$  a diodou v optočlenu nebo část opačná, s tranzistorem optočlenu a rezistorem  $R_e$  (Obr. 6).

Směr Raspberry Pi - > optočlenu připojené na sběrnici:

$$U_H - \text{napětí vysoké úrovně} = 3,3V - 3,5V$$

$$U_{\text{Napájecí}} - \text{napájecí napětí} = 3,3\text{V} - 3,5\text{V}$$

$I_B$  – proud bázevým rezistorem - nejlépe < 1mA (jak pro GPIO tak pro A/D převodník)

$$I_f - \text{proud diodou optočlenu} = 1\text{mA} - 50\text{mA}, \text{ zvoleno } 5\text{mA}$$

$$I_D - \text{proud tranzistorem optočlenu} < 30\text{mA}$$

$$U_F - \text{Napětí na diodě optočlenu} = 1,1\text{V}$$

$$U_{CE} - \text{napětí sepnutého tranzistoru optočlenu} = 0,7\text{V}$$

$$U_b - \text{napětí báze tranzistoru při průchodu proudem} - 0,6\text{V}$$

Z důvodu minimalizace počtu různých hodnot jsem navrhl hodnoty shodné pro obě napájecí napětí. Navíc se obě napájecí napětí příliš neliší, tudíž by hodnoty vyšly tak, že by stejně obě odpovídaly stejné reálné součástce.

#### Vstupní část optočlenu:

Rezistory v bázi tranzistoru:

$$R_b = \frac{U_H - U_b}{I_b} = \frac{3,5 - 0,6}{1 \cdot 10^{-3}} = 2,9\text{ k}\Omega \quad (2.6)$$

byly zvoleny rezistory hodnoty 4,7kΩ; skutečný proud je :

$$I_b = \frac{U_{Rb}}{R_b} = \frac{3,5 - 0,6}{4700} = 0,6\text{ mA} \quad (2.7)$$

Rezistory omezující proud diodou optočlenu:

$$R_f = \frac{U_{\text{Napájecí}} - U_F}{I_F} = \frac{3,5 - 1,1}{5 \cdot 10^{-3}} = 480\ \Omega \quad (2.8)$$

byly zvoleny rezistory hodnoty 470 Ω; skutečný proud je :

$$I_b = \frac{U_F}{R_F} = \frac{3,5 - 1,1}{470} = 5,1\text{ mA} \quad (2.9)$$

Rezistory omezující proud výstupní část optočlenu:

$$R_e = \frac{U_{\text{Napájecí}} - U_{CE}}{I_b} = \frac{3,5 - 0,7}{1 \cdot 10^{-3}} = 2,8\text{ k}\Omega \quad (2.10)$$

byly zvoleny rezistory hodnoty 4,7kΩ; skutečný proud je :

$$I_b = \frac{U_{Rb}}{R_b} = \frac{3,5 - 0,7}{4700} = 0,59 \text{ mA} \quad (2.11)$$

## 2.3 DC – DC měnič

Návrh hodnot měniče vycházel z volnějších požadavků na přenášený výkon a účinnost. K předběžnému návrhu počtu závitů jsem využil softwarového návrhového nástroje od ON Semiconductor, což je jeden z výrobců obvodu typu 34063, který je použit v tomto měniči. Softwarový nástroj jako vstupní data požaduje velikost jádra, maximální sycení, přenášený výkon, účinnost, vstupní a výstupní napětí a střídu. Jedná se o návrh počítající s přenosem zadaného výkonu, to znamená při zatížení.

**FLYBACK XFMR DESIGN**

DATA INPUT		RESULTS	
Peak output power (Wpk)	0,08	Peak pri current =	0,2222222222
Min dc input voltage (on Cbu)	1,6	Primary rms current =	0,095149871
Efficiency estimate (dec. #)	0,9	Primary L (μH) =	80
Switching freq (kHz) (set to 45 kHz for QR design)	45	Primary turns =	8,2304526749
Core area (Ae in sq cm)	0,18	Min secondary turns =	14,7306397306
Max core flux (kilogauss)	1,2	Approx core gap-inches =	0,0009144947
Output voltage	3,5		
Duty Cycle max. (set to 0.6 for QR design)	0,55		

**= Enter Data**                      **= Calculated Data**

Obr. 10 Snímek z návrhového nástroje od společnosti ON Semiconductor

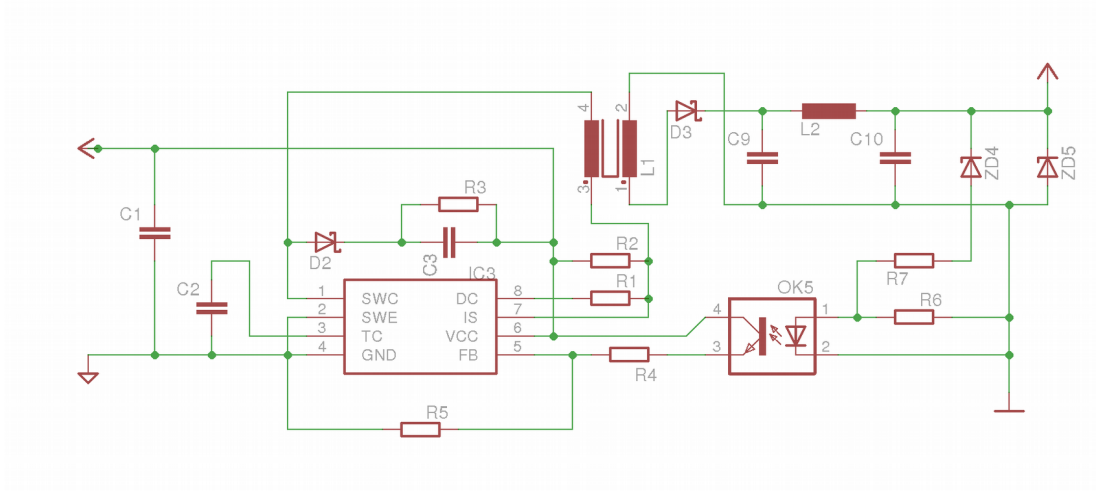
V mém případě měnič bude pracovat v tzv. „Burst“ režimu, který lze nazvat jako střídavě relaxační. Do tohoto stavu se dostane měnič ve chvíli, kdy požadavek na přenášený výkon je tak malý, že je mimo schopnosti regulace pomocí změny střídy. Napětí na výstupu měniče i s nejmenší možnou střídou plnění cívky vzroste nad určenou úroveň 3,5V a celý měnič se na okamžik než výstupní napětí klesne pod úroveň 3,5V zablokuje. Poměr časů, kdy je měnič v provozu a kdy je zablokovaný, je u provozu bez zátěže velký, přičemž je měnič po většinu doby vypnut a pouští jen krátké série impulzů – tzv. „Bursty“. Bez připojené zátěže

je měnič zatížen jen proudem tekoucím zpětnovazebním optočlenem, který jsem při volné regulaci naměřil přibližně 0,5 mA.

Výpočet snímacího rezistoru  $R_2$  :

$$R_2 = \frac{0,3}{2 \cdot I_{\text{výstupní}} \left( \frac{t_{\text{zapnuto}}}{t_{\text{vypnuto}}} + 1 \right)} = \frac{0,3}{2 \cdot 8 \cdot 10^{-3} \left( \frac{1}{1} + 1 \right)} = 9,375 \Omega \quad (2.12)$$

Byl použit rezistor 10  $\Omega$ . Časy  $t_{\text{zapnuto}}$  a  $t_{\text{vypnuto}}$  jsem zvolil o hodnotě 1, neboť jsou v podílu a smyslem je vyjádřit střídání spínání proudu 50 % a  $I_{\text{výstupní}}$  značí maximální předpokládaný výstupní proud.



Obr. 11 DC – DC měnič s obvodem MC34063

Rezistor  $R_1$  omezuje spínací proud do báze tranzistoru. Jeho hodnota by měla být dost malá aby nedocházelo ke zbytečným úbytkům na výstupním tranzistoru, vlivem nedostatečného otevření tranzistoru, ale zase příliš nízká hodnota znamená zvýšený proudový odběr.

$$R_1 < 100 \cdot U_{\text{napájecí}} - 100 \quad (2.13)$$

a zároveň

$$R_1 < \frac{50 \cdot U_{\text{napájecí}}}{\left( \frac{0,3}{R_3} \right)} \quad (2.14)$$

První podmínce vyhovuje hodnota do 230  $\Omega$ . Druhá podmínka vychází při malém výkonu v řádu k $\Omega$  a neuplatní se tedy. Byla zvolena hodnota 180  $\Omega$ .

## 3. NÁVRH SOFTWAREVÉ ČÁSTI

Pro řízení a záznam z měřících modulů jsem zvolil jednodeskový mikropočítač Raspberry Pi, dodán byl v současné době v jeho poslední verzi 3, model B. Založen je na procesoru Broadcom BCM2837 je kategorie System-on-Chip, tudíž sdružuje velké množství funkcí do sebe. Například vstupně výstupní piny GPIO jsou vyvedeny přímo z tohoto procesoru nejsou odděleny žádným pomocným vstupně-výstupním obvodem, s čím ale také souvisí mnohem větší opatrnost při manipulaci a ožívání. Raspberry Pi tak dostává rozměry jen o něco málo větší, než je platební karta. Procesor je čtyř-jádrový o taktu 1.2GHz, s adresní sběrnici o šíři 64-bitů a operační paměť RAM o velikosti 1 GB.

Vstupně-výstupní piny na GPIO konektoru jsou softwarově nastavitelné. Nastavení se provádí zápisem do registru. O mechanismu nastavování proměnných v procesoru se není třeba více rozepisovat, protože pro účely mé práce je postačující nastavení vstupně-výstupní funkce a hodnoty zprostředkované skrz aplikační rozhraní systému.

### 3.1 Popis operačního systému

Na microSD kartě v mikropočítači je nainstalován Linux, konkrétně distribuce Raspbian, určená pro Raspberry a v odlehčené variantě Lite, která neobsahuje grafické aplikace. GNU/Linux je svobodný a otevřený operační systém z rodiny unixových systémů, podporující drtivou většinu dnešních procesorů. Tato vlastnost společně se stabilitou a velkou modularitou pomohla rozšíření systému Linux do velkého množství jak domácích, tak průmyslových nebo komerčních zařízení.

#### 3.1.1 Jádro a adresářová struktura

Jádro Linuxu je monolitické, avšak s možností zavádění modulů. Jádro se nahrává do paměti po zapnutí napájení a předání od firmware zařízení. Po zavedení jádra se načítají moduly s ovladači zařízení, procesy a další služby, jako například připojení souborových systémů, obsluha sítě, a další.

Adresářová struktura nezačíná písmenem jednotky, jak je zvykem z MS Windows nebo dříve z MS-DOSu, ale začíná lomítkem.

*/ - kořenový adresář*

*/bin - spustitelné soubory ( stejně jako /sbin, /usr/bin, /usr/sbin/ )*

*/boot - soubory zavaděče*

*/dev - soubory zastupující zařízení (virtuální soubory – nejsou fyzicky na médiu )*

*/etc - nastavení a konfigurační soubory*

*/home - domovské složky uživatelů*

*/lib - sdílené knihovny a ovladače zařízení (stejně jako /usr/lib/)*

*/mnt - přípojný bod diskových oddílů (stejně jako /media )*

*/proc - soubory zastupující procesy (virtuální soubory – nejsou fyzicky na médiu )*

*/root - domovská složka správce*

*/run - soubory bezprostředně související s během procesu ( dříve /var/run/ )*

*/tmp - dočasné soubory*

*/sys - soubory zastupující systémové proměnné a stavy registrů*

*/var - obsahuje soubory a složky dočasnějšího charakteru (například zápisy systému)*

Jak je patrné, vše má své místo. Všechna vstupní i výstupní zařízení, procesy, diskové svazky apod. jsou v Linuxu vedena někde jako soubor. To dává možnost do něj zapisovat pomocí standardních nástrojů o kterých se více rozepíšu v kapitole 2.1.4 BASH. Kde tyto soubory hledat, je také velmi dobře zdokumentováno v dokumentaci, která se nachází přímo v systému a bývá obvyklou součástí každého instalovaného programového balíčku.

Balíčkový systém je záležitost, která je v prostředí Linuxu již pevně zakotvena. Uživatel systému GNU/Linux má dnes prakticky dvě hlavní cesty, jak se dostat k softwaru, který chce. Varianta první je, že si někde opatří zdrojový kód, obvykle zabalený v archivu .tar.gz s patřičnými instrukcemi a obvykle i sadou skriptů, které provedou hromadné kroky kompilace za uživatele. Varianta druhá spočívá v provozování nějakého ze systémů pro správu již předkompilovaných balíčků, kde spustitelné soubory jsou již v binárním formátu, ale bývají přítomny informace, ze kterých instalační program balíčkovací služby provede dodatečnou konfiguraci podle potřeb systému. Balíčkovací systém má obvykle vyřešeno i ověření pravosti instalovaného balíčku. Varianta známá z prostředí operačního systému Windows, kdy je uživatel nucen spouštět neznámý binární soubor, o kterém až do dokončení instalace nemusí mít podrobné informace ani ověření že je pravý, který navíc musí v té dané verzi systému, se zrovna přítomnými knihovnami být spustitelný, kde tento program teprve instaluje další programy a soubory, se v prostředí Linuxu používá jen málo a obvykle u komerčních programů, které ať už z licenčních důvodů nebo jiných neposkytnou svůj

program ve formě balíčku do partnerských repositářů Linuxových distribucí tak, jak to dělá většina velkých softwarových firem vyvíjejících software pro využití širokou řadou uživatelů. Raspbian využívá balíčkový systém apt.

Další podstatnou vlastností je systém práv. Vzhledem k tomu, že Linux byl od počátku koncipován jako víceuživatelský systém, bylo nutné zavést systém práv. Ten vychází z Unixu, je poměrně jednoduchý, leč s pár přidanými parametry stále dodnes plní vše co je zapotřebí. Každý soubor je někým vlastněn, pokud ne, dědí oprávnění adresáře. Oprávnění se zapisují znakem *r* pro čtení, *w* pro zápis, *x* pro spuštění a *-* je bez oprávnění. Zapisují se ve třech trojicích protože se mohou vztahovat na uživatele, skupinu nebo na ostatní. Možný zápis pak vypadá *rw- rw- r--* což značí že soubor je pro čtení a zápis pro uživatele, který je vlastníkem souboru ten jej může také spustit, pro skupinu ve které je uživatel je soubor ke čtení a k zápisu a pro ostatní uživatele a skupiny je pouze ke čtení. Případně je možný i zápis třemi čísly 0-7, z nichž každé odpovídá stavům interpretovaným jako bity v BCD hodnotě, předchozí stav by se tedy zapsal číslem 764 (7=*rw*x, 6=*rw-*, 4=*r--*).

Nad všemi uživateli je jeden super-uživatel, nazývá se root a má práva plného přístupu všude. Jeho domovský adresář se na rozdíl od ostatních uživatelů nenachází v */home/jméno* ale v */root*. Není to účet pro běžnou činnost, ale slouží výhradně ke správě systému. Pokud běžný uživatel chce spustit nějaký program, či příkaz vyžadující práva super-uživatele root, je možné si o ně dočasně zažádat připsáním příkazu *sudo* nebo ekvivalentního před spuštěným příkazem, pokud je uživatel zapsán v souboru */etc/sudoers* jsou práva příkazu přidělena.

Práva nastavuje vlastník souboru příkazem *chmod*, kde *chmod 777 soubor* nastaví souboru práva kdy všichni mohou soubor číst, zapisovat do něj a spouštět. Vlastníka souboru změním příkazem *chown*.

### 3.1.2 Souborové systémy

Operační systémy založené na jádru GNU/Linux podporují obvykle širokou řadu souborových systémů. Pro kořenový adresář se dnes na osobních počítačích nejčastěji používá souborový systém Ext3 nebo Ext4. Starší varianta Ext2 je stále podporovaná, využívaná už v čím dál menší míře neboť nepodporuje žurnálování. Tato vlastnost je výhodná v případě pádu systému, neboť před provedením samotného zápisu se u žurnálování metadat nejdříve do datové oblasti zvané žurnál zapíše nejdříve co se kde bude zapisovat a pak se teprve

provede samotný zápis a až po úspěšném zápisu, se záznam v žurnálu odstraní. Žurnálování celých dat se běžně nevyužívá z důvodu delšího zápisu. Tento mechanismus pomáhá udržet konzistenci dat i při náhodném pádu systému a zároveň urychluje kontrolu disku, neboť není potřeba kontrolovat celý disk, konkrétně lze při kontrole po nevhodném vypnutí systému kontrolovat a opravovat pouze chyby vzniklé při posledním zápisu.

Po startu systému se připojují souborové systémy podle nastavení uloženého v souboru `/etc/fstab` a `/etc/mstab` za běhu systému obsahuje právě připojené svazky.

Ukázka souboru `/etc/fstab` s vysvětlivkou v prvním řádku:

<u>Co se připojuje?</u>	<u>Kam?</u>	<u>Typ</u>	<u>Volby</u>	<u>záloha</u>	<u>kontrola po startu</u>
<code>/dev/mmcbk0p1</code>	<code>/boot</code>	<code>vfat</code>	<code>defaults</code>	<code>0</code>	<code>2</code>
<code>/dev/mmcbk0p2</code>	<code>/</code>	<code>ext4</code>	<code>noatime,nodiratime,commit=120</code>	<code>0</code>	<code>1</code>
<code>tmpfs</code>	<code>/tmp</code>	<code>tmpfs</code>	<code>defaults,nodev,nosuid</code>	<code>0</code>	<code>0</code>
<code>/dev/mmcbk0p3</code>	<code>/log-part</code>	<code>vfat</code>	<code>defaults</code>	<code>0</code>	<code>0</code>

Tato ukázka je použita přímo z této práce, proto v prvním sloupci je jako zdroj uvedeno zařízení např. `/dev/mmcbk0p2`, což je o označení druhé oblasti na první přítomné paměťové kartě (MMC je předchůdce technologie SD karet). Pevné disky se obvykle označují `/dev/hda2` nebo `/dev/hdb1` kde písmeno za `hd` označuje pořadí disku a číslo pořadí svazku na disku, nověji však u disků připojených přes sériové rozhraní, které je dnes standardem, se obvykle vyskytují jako `/dev/sda1` nebo `/dev/sdb2`.

Pro správu oddílů lze využít program `fdisk`. Tento program nabízí širší podporu činností, typů systémů souborů, nastavení a užitečných informací, než jeho DOSový protějšek stejného názvu.

Kontrolu svazku zajišťuje program `fsck`, ten je spouštěn podle pravidel občas nebo pokaždé při startu systému a provede kontrolu pokud je zapotřebí. Kontrolu lze u některých souborových systémů vynutit přepínačem `-f`.

Pro běh systému na takovém médiu jako je paměťová karta, je zapotřebí brát na zřetel základní vlastnosti tohoto média. Mezi nimi zvláštní pozornost patří například omezený počet



přepisů který se pohybuje u lepších paměťových karet kolem 100 000 zápisů, poté dochází ke ztrátě funkce. Operační systém stále provádí nějaký zápis na disk, obvykle se jedná o zápis souborů se záznamy nebo u Raspberry také ukládání posledního stavu hodin do souboru *fake-hwclock*, odkud je čas po startu zaveden a následuje pokus o zavedení skutečného času ze sítě.

Nevhodným programem, který bude například z důvodu chyby stále přepisovat jedno místo mnohokrát za vteřinu, lze paměťovou kartu zničit během několika minut. Běžná životnost dnešních SD karet je dle zkušeností půl až jeden rok, pokud není systém přizpůsoben pro snížení počtu zápisů.

Tohoto snížení lze dosáhnout vhodným zápisem v souboru */etc/fstab*, který spočívá v připojení všech umístění se soubory dočasného charakteru, které není potřeba uchovávat do souborového systému vytvořeného v paměti RAM namísto jejich zápisu na SD kartu. Parametrem *noatime* zabráníme přepisování času posledního použití souboru, to také šetří počet přepisů. Připojením souborového systému ext3 jako ext2 způsobíme, že se nebude používat žurnálování. U ext4 se tato možnost provádí utilitou *tune4fs*.

Všechny tyto metody vedou jen ke snížení počtu zápisů provedených operačním systémem. Nejlepší volbou by bylo připojovat kořenový systém pouze pro čtení a zápisy navíc zcela eliminovat. Pro zápis připojit jen vytvořenou oblast na paměťové kartě pro naměřená data a ta navíc nezaznamenávat hned po jejich naměření ale ukládat je například do operační paměti a na paměťovou kartu nakopírovat až větší množství dat najednou. Vzhledem k tomu, že převod operačního systému do takové podoby aby mohl běžet v režimu pouze ke čtení je komplexnější záležitost, nebylo toto definitivní řešení prozatím aplikováno. Možné je také využít speciální distribuce, určené pro běh z nezapisovatelného souborového systému, které mají již vše potřebné vyřešeno

### 3.1.3 Služby

Obvyklou součástí systému jsou různé ať už uživatelské či systémové služby. Správu těchto služeb, jejich spouštění při startu a ukončování při vypínání systému vykonává správce služeb. Dříve bylo běžné, že se služby manuálně spouštěly přes skripty v adresáři */etc/init.d/* a jejich spouštění při startu zajišťoval systémová služba *init*. V současné době je zvyklostí používat k manuálnímu spouštění, zastavování a restartování služeb program *service*

a spouštění služeb při spouštění systému se na desktopových verzích zajišťuje dnes obvykle služba `upstart`.

Spuštění služby provedeme příkazem

```
service start název
```

Další parametry jsou `stop`, který službu zastaví a `restart`, který ji zastaví a znovu spustí.

```
service --status-all slouží pro výpis služeb a označení spuštěných služeb.
```

```
service lighttpd restart způsobí vypnutí a opětovné zapnutí služby webového serveru
```

### 3.1.4 BASH

Je interpret příkazové řádky, zvaný také shell. Umožňuje uživateli se do systému přihlásit, odhlásit se z něj, spouštět programy a sledovat výstup z nich. Vytváří tak textové uživatelské rozhraní, které poskytuje vše potřebné pro ovládání systému. Umožňuje také spouštění dávkových úloh. Před samotným spuštěním ještě shell provádí některé úpravy – interpretace, například dosazuje za proměnné a za zástupné znaky v názvech souborů. Interpret není vše, tudíž jsou součástí bash také doprovodné programy - utility, z nichž každá něco dělá. Filozofie systémů z rodiny unix je nedělat kus kódu dvakrát a topologie vznikala v dobách mnohem skromnějších na výpočetní výkon i velikost operační paměti, dnes tedy tyto komponenty obvykle pracují tak rychle, že je obvyklá i situace, že vlastní napsaný kód být na míru dané aplikaci nemusí vždy poskytovat výsledek rychleji, než příkaz na první pohled velmi komplexní, u kterého by se dalo očekávat, že je určen spíše pro uživatelské vstupy, než na automatizovanou činnost. Vhodným propojením těchto standardních systémových utilit v dávkovém souboru, který podporuje proměnné, podmínky, cykly a další záležitosti, jež jsou spíše doménou programovacích jazyků. Je častá situace, že i uživatelé, kteří nejsou primárně programátory, ale spíš pasivní uživatelé Linuxu na osobním počítači, si po čase všimnou, že v jejich systému je příkazová řádka nabízející netušené možnosti a navíc pokud již příkazy známe, tak i velmi rychle dostáváme výsledek, neboť nejsme zdržováni občas nevhodným ovládáním programů pomocí myši. A pokud jsou známy příkazy k vykonání, zapsat je do souboru shellového skriptu, který pak jde uživatelsky spustit, už není obtížné.

Důležitou funkcí je řetězení a přesměrování výstupů příkazů, kdy je možné výstup z jednoho příkazu přivést na vstup jiného. Toto přesměrování probíhá na systémové úrovni,

tudíž s vysokou rychlostí a propustností. Je takto možné přesměrovávat i datové proudy, například videosnímek, který je možné jedním příkazem současně stahovat, během stahování souboru jej například konvertovat do jiného formátu a tento přehrávat.

Přesměrování výstupu do souboru lze provést připsáním znaménka větší-menší :

```
echo '1' > /sys/class/gpio/gpio20/value
```

Tento příkaz vloží hodnotu 1 do souboru, předchozí obsah se maže, tento konkrétní příklad ukazuje zápis logické 1 do souboru zastupující danou vlastnost zařízení, tím i zapnutí pinu 20 na GPIO konektoru.

```
date >> soubor.txt
```

Tento příkaz přidá aktuální datum a čas na konec souboru *soubor.txt* . Pokud spustíme vícekrát, bude se dále přidávat další řádek. Takto vytvořený soubor můžeme vypsát pomocí příkazu *cat*:

```
cat soubor.txt
```

```
Čt kvě 20 16:46:42 CEST 2017
```

```
Čt kvě 20 16:46:42 CEST 2017
```

```
Čt kvě 20 16:46:42 CEST 2017
```

Příkazy se obvykle provádí s parametry, ty se zapisují za jednoduchou pomlčku jednoznačně *příkaz -a -v*, nebo řetězcem *--all --verbose*. Jednoznakové se dají seskupovat *-av* . Pokud má být parametrem několik souborů, lze využít zástupný znak \* - hvězdička, který značí, všechny znaky od tohoto znaku dál. Například *rm \*.jpg* smaže všechny soubory s příponou jpg.

Pro práci se soubory a složkami je vhodné znát několik příkazů:

*cd* – change directory – změna adresáře ; '*cd ..*' slouží k posunu o úroveň výš

*ls* – výpis adresáře; '*ls -h*' vypíše velikost v násobcích, namísto v jednotkách bytů

*mkdir* – vytvoření adresáře

`rmdir` – smazání adresáře, implicitně maže pouze prázdné adresáře

`rm` – mazání souborů, s parametrem `-rf` maže soubory i adresáře rekurzivně

`cp` – kopírování souborů

`cat` – slouží ke spojování a k výpisu souborů na standardní výstup;  
*cat souborA souborB souborC > souborABC*

`dd` – kopírování souboru znak po znaku; *dd if=obraz.img of=/dev/sdc* – nahraje soubor `obraz.img` obsahující obraz disku na `/dev/sdc` – což může být například USB flash disk

`grep` - výpis řádků obsahujících řetězec;

*dmesg | grep - 'IP'* vypíše ze systémového logu řádky začínající řetězcem 'IP'

`tail` – podobně jako `grep` formátuje výstup – `tail` vypisuje posledních 10 řádků

`sed` – pokročilejší formátování textového proudu pomocí pravidel z formátovacích parametrů

`/dev/null` – zvláštní znakové zařízení, které vše co se do něj zapíše ze ztratí ale zápis proběhne úspěšně, někdy je potřeba textový výstup nezobrazit ale zahodit jej.

*dmesg | tail >> /dev/null*

Tento příkaz přečte ze systémového logu posledních deset řádků a ty pak zahodí. Znak `|` značí tzv. rouru, která slouží k přesměrování výstupu na vstup jiného procesu.

`gcc` – překladač jazyka C

`g++` – překladač jazyka C++

`nano` – editor souborů

`man` - velmi užitečný manuál k většině příkazů a k velkému množství konfiguračních souborů, například *man fstab* vypíše nápovědu k editaci souboru `/etc/fstab`, který jsem popisoval v části 2.1.2 Souborové systémy.

Zde uvedený výčet je zkrácený a rozhodně nepokrývá všechny příkazy. Spoustu dalších příkazů lze zobrazit příkazem *help* – Vyvolá nápovědu pro BASH.

Užitečným pomocníkem je v příkazové řádce doplňování příkazů, cest a někdy i parametrů příkazů, pomocí klávesy TAB. Ve spojení s historií příkazů přístupnou pod klávesami šipka

nahoru/dolu, získává práce v příkazové řádce zcela jiný rozměr, než jsou uživatelé zvyklí například z prostředí DOS. Rychlost práce je u uživatelů zvyklých na ovládání moderních shellů všeobecně podobná, ba dokonce v některých případech předčí grafické rozhraní. Jsou samozřejmě činnosti, kde je grafické prostředí nevyhnutelné, pro aplikace serverové povahy se grafické prostředí obvykle ani neinstaluje, pro zkušeného administrátora je příkazová řádka rychlejší a pohodlnější, zvláště pro hromadné úlohy, kde je obvykle grafické prostředí bez pomocných nástrojů nepoužitelné a stejně uživatel pouští terminál.

Mikropočítač Raspberry Pi má na portu GPIO také piny sériového rozhraní typu UART s napěťovými úrovněmi 3,3V a poskytuje na nich ve výchozím nastavení přístup k sériové konzole o rychlosti 115200 Bd. Toho se dá využít pro případ, kdy by Raspberry přestalo komunikovat po síti Ethernet nebo Wi-Fi a bylo by zapotřebí provést nějakou činnost, která nesnese tvrdý restart odpojením zdroje, neboť by došlo ke ztrátě neuložených dat.

### 3.1.5 Síť a nastavení času

Pro funkci zařízení, které zaznamenává jakékoliv informace v čase, je důležité aby byl čas řádně nastaven. Raspberry Pi nemá integrované hodiny reálného času, proto se musí o systémový čas postarat sám systém. Standardně se k tomu u systému Raspbian využívá *fake-hwclock*, což je simulace hardwarových hodin, kdy se pokračuje v čase, který byl před vypnutím. Hodiny jdou sice špatně, leckdy je ale přednější udržet kontinuitu dat a časový sled záznamů a to nejen uživatelských dat, ale i systémových. Při startu systému se po navázání spojení se sítí synchronizují hodiny pomocí síťového protokolu NTP – Network Time Protocol..

Kromě toho, je síťové rozhraní použito k ovládání a správě systému přes zabezpečené rozhraní SSH – SecureShell. Další funkcí pro síťové rozhraní je provoz webového serveru s naměřenými údaji a jejich grafickou interpretací. Takto je možné data zobrazit kdekoliv z lokální sítě, kde bude měřicí systém umístěn.

Pro spojení s počítačovou sítí lze využít port Ethernet nebo nově také bezdrátové připojení Wi-Fi. Správa síťových rozhraní se děje pomocí příkazu *ifconfig* nověji *ip* a vlastnosti bezdrátových rozhraní se děje pomocí příkazu *iwconfig*.

## 3.2 Popis ostatních komponent

Kromě základního systému jsou na microSD kartě nainstalovány další komponenty potřebné pro střádání a zobrazení dat. Správu naměřených údajů a vykreslení grafů zajišťuje programový balík rrdtool a výsledná data jsou zobrazována na webové stránce, jejichž publikování zajišťuje webový server lighttpd.

### 3.2.1 Správa databáze RRDTool

Sada nástrojů RRDTool je určena pro správu databáze dat přímo závislých na času. Jedná se o komplexní sadu pro vytváření, vkládání, výpis a grafický výstup hodnot vkládaných v určitý čas. Nástroj sám zvládá několik rozdílných zdrojů dat od prosté hodnoty po vyjádření například rychlosti síťového rozhraní a přičítání do hodnoty množství přenesených dat.

Databáze je typu round-robin, což značí její funkci a vlastnosti. Nástroj vytvoří patřičně velký soubor podle zadaného počtu uchovávaných hodnot a pokud je databáze zaplněna, sama se postará o zahození starých hodnot. Databáze tak nepřeroste zadanou velikost souborů.

Pro vytvoření databáze nám poslouží volba *rrdcreate*. Zde se nastaví povaha zdroje dat DS, která může být více typů:

GAUGE- vyjádření prosté hodnoty, například teplota nebo napětí

COUNTER - počítadlo, údaj, který stále roste, například množství přenesených dat

DERIVE – počítadlo, které může i klesat, pokud je vstup záporný

ABSOLUTE – údaj je nejdříve potřeba vztáhnout na čas pro získání prosté hodnoty

Dále se pomocí RRA nastaví vlastnosti pomyslných sloupců v tabulce, jejich délka, rozsah přijatelných hodnot, kde hodnoty mimo rozsah nejsou započítány, interval přibývání hodnot v databázi a funkce uplatněná na hodnoty. Funkce může být :

AVERAGE – průměrná hodnota

MIN – nejnižší hodnota

MAX – nejvyšší hodnota

Nástroj *rrdupdate* se stará o přidávání hodnot do databáze. Zápis vypadá takto:

*rrdupdate soubordatabáze.rrd N:1:2*

Příkaz provede zapsání hodnoty 1 do prvního sloupce a 2 do druhého sloupce, s časovou značkou odpovídající času spuštění tohoto příkazu, pro jiný čas se místo N zapíše časová značka.

Vytvoření souboru s grafem naměřených hodnot je zapotřebí spustit *rrdtool graph* s odpovídajícími parametry:

```
rrdtool graph soubordatabaze.rrd --start [čas začátku grafu] --end [konec grafu, pokud není  
zadáno tak teď] --step [interval mezi kroky] DEF,VDEF,CDEF:[definice datové oblasti]  
LINE,AREA,GPRINT,PRINT: [definice elementů zobrazovaných v grafu]
```

Nástroje obsažené v RRDTool jsou velmi obsáhlé co se možností týká, je tedy možné grafický výstup přizpůsobit.

### **3.2.2 Webový server Lighttpd**

Pro zobrazení naměřených hodnot přes webové rozhraní je potřeba HTTP server. Z důvodů menších nároků na systémové prostředky jsem vybral software Lighttpd. Tento webový server je rychlý, bezpečný a přizpůsobitelný nejružnějším požadavkům. Konfigurace serveru je umístěna v souboru */etc/lighttpd/lighttpd.conf* a po provedení změn je vhodné provést restart služby webového serveru příkazem

```
service lighttpd restart
```

Soubory určené k publikování jsou v základním nastavení umístěny ve složce */var/www* a první soubor, který se načítá, je *index.html*.

Pro psaní webových stránek se používá značkovací jazyk HTML. V tomto jazyku se píše formátovací značky – HTML tagy. Tyto jsou dvojího druhu, párové a nepárové. Tag párový má tag počáteční například *<html>*, mezi kterým je obsah a tag koncový *</html>*, který obsah uzavírá. Tagy nepárové, například *<br>* pro další řádek se píše samostatně tak , jak jsou.

*<html>* tag uvozující html stránku

*<head><title>Titulek stránky</title></head>* hlavička s titulkem stránky

*<body>* tělo stránky

*<h1> Nadpis </h1>*

*<p> Odstavec textu</p>*

`<a href="stranka.html"> Odkaz </a>`

`<br>` - další řádek

``

`</body>` tag uzavírající obsah stránky

`</html>` - tag uzavírající celý HTML soubor

### 3.3 Sběrnice a komunikace s okolím

Mikropočítač Raspberry umožňuje komunikaci hned po několika sběrnících sériového charakteru. V této kapitole se budu zabývat sběrnicí SPI, která je použita v mé práci pro načítání hodnot z analogově-digitálních převodníků a sběrnici 1-Wire, která je použita k připojení teplotních čidel DS18B20.

#### 3.3.1 Sběrnice SPI a komunikace s A/D převodníky

Jedná se o poměrně jednoduchou synchronní sériovou sběrnici. Na sběrnici je přítomen jeden nadřazený uzel – Master, který řídí uzly podružné – Slave. Jen Master je oprávněn řídit a zahajovat přenos. Pro obousměrnou komunikaci je potřeba tři datových vodičů, data vstupní, data výstupní a hodinový vstup. Dále je zapotřebí jeden vodič pro každé zařízení na sběrnici. Tento vodič výběru protějšku – Chip select, nabývá stavu logické 0 nebo logické 1, podle toho, jestli se zařízením zrovna probíhá přenos. Polarita bývá obvykle invertovaná, to znamená, že pokud chceme například s analogově digitálním převodníkem MCP3202 zahájit komunikaci, je potřeba nejdříve pin /CS spojit se zemí, tj přivést na něj potenciál logické nuly.

Obsluha je možná jak zápisem přímo do zařízení příkazem

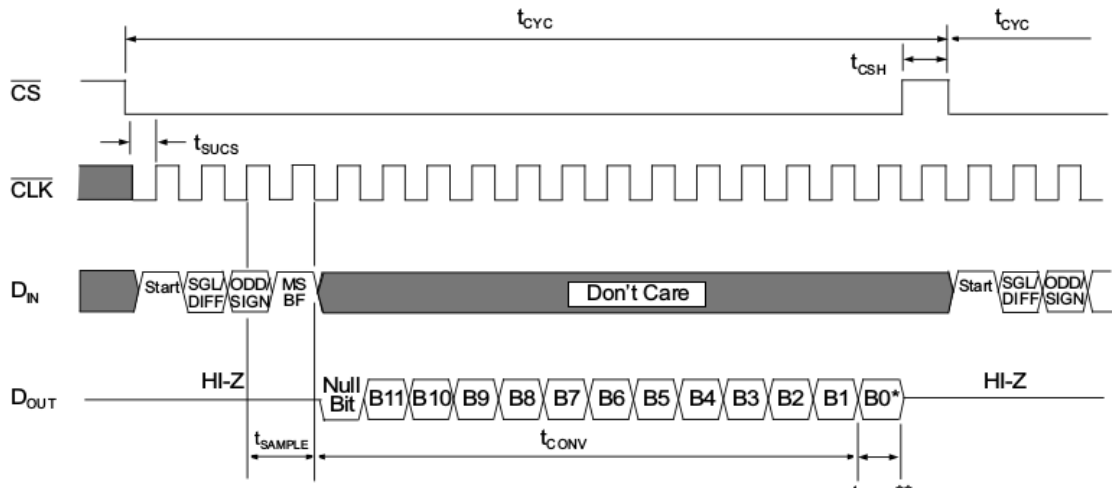
```
echo -ne 'řetězec' > /dev/spidev0.0
```

Takový přístup je pouze pro zápis. Pokud chceme kompletní komunikaci, je vhodnější zvolit program využívající systémovou knihovnu. V mé práci jsou pro tuto činnost využity dva spustitelné soubory m0, který měří první kanál a m1, který měří druhý kanál zrovna vybraného analogově-digitálního převodníku. Po každém ukončení dochází vlivem překlopení signálu /CS k přepnutí výstupu čítače s výstupem 1 z 10 typu 4017 na další vodič, tím pádem je zvolen další obvod.

Obvod dekódování adresy 4017 má z řídicího mikropočítače ovládaný také pin



nulování-reset. Pokud by dekodér adresy osazen nebyl, bylo by nutné využít 8 samostatných vodičů z GPIO konektoru pro přepínání zrovna aktivní měřící desky. Pro obsluhu tohoto pinu byl napsán jednoduchý skript *reset.sh*, který provede vynulování krátkým sepnutím určeného pinu na konektoru GPIO.



Obr. 12 Komunikace s převodníkem MCP3202

### 3.3.2 Sběrnice 1-Wire a měření teploty

Vyčítání teploty z teplotních čidel DS18B20 probíhá po sběrnici 1-Wire. Obsluha této sběrnice probíhá skrze ovladač v jádře. Sběrnice 1-Wire, jak už název napovídá, využívá pouze jednu datovou linku, přičemž obvod umožňuje i napájení z této linky, čímž by šlo použít místo tří vodičů, vodiče dva. Po povolení sběrnice 1-Wire v konfiguraci, systém sám začne sbírat potřebná data, která zpřístupní, jako soubor, odkud jej lze vyčíst například pomocí nástroje *cat*. Čidlo poskytuje hodnotu jako číslo v tisícinách stupně.

Každé zařízení 1-Wire, má přiděleno unikátní 64-bitový identifikátor a k jedné sběrnici lze připojit až 75 zařízení. Začátek tohoto identifikátoru značí druh zařízení. Sběrnice musí být zakončena pull-up rezistorem, který je pro všechny zařízení na lince společný, Má běžně hodnotu  $4,7k\Omega$  a v případě chyb při přenosu ze vzdálenějších čidel jej lze zmenšit.

## 4. REALIZACE SOFTWAREVÉ ČÁSTI

Obslužný algoritmus je zapsán jako skript v jazyce BASH. Název souboru je *writelog.sh*

a byl napsán v editoru *nano*, přímo na Raspberry Pi. Komunikace počítače a Raspberry Pi probíhá přes zabezpečený terminál SSH, tudíž je možné ladit software zařízení rovnou za běhu.

Celý program je umístěn na FAT 32 oblasti SD karty */log-part*, kam se posléze ukládají data. Po startu se celý program zkopíruje do pracovního adresáře v dočasném umístění */tmp/workdir*.

Vzhledem k tomu, že celé umístění */tmp* je zápisem v souboru */etc/fstab* připojeno do virtuálního souborového systému *ramfs*, dojde při vypnutí napájení ke ztrátě těchto dat. Pro omezení přístupů na kartu jsem zvolil řešení, kdy se po určité době naměřená data zapíše na kartu. Jedná se o příkaz *cp* pro zkopírování souboru *database.rrd* a *data.csv* na paměťovou kartu. Příkaz jsem zapsal do časovací služby *cron*. Konkrétně do souboru umístěného v adresáři */etc/cron.hourly*, jehož obsah služba *cron* spouští každou hodinu. Výpis načasovaných úloh zkontrolujeme s příkazem: *crontab -e*

## 4.1 Spuštění

Je žádoucí aby datalogger zahájil zápis naměřených údajů do souboru samostatně po připojení k napájení. K tomuto účelu jsem přidal řádek ve startovacím skriptu */etc/rc.local*

```
/root/./start.sh &
```

Kde *start.sh* je skript který má za úkol zkopírovat pracovní soubory z FAT oblasti z karty, která je připojena do */log-part*, do dočasného pracovního adresáře v */tmp/workdir*

```
#!/bin/bash - hlavička BASH skriptu  
mkdir /tmp/workdir - vytvoří pracovní adresář  
cp /log-part/_WD/* /tmp/workdir/ - zkopíruje do něj soubory z SD karty  
cd /tmp/workdir - přejde do adresáře  
./writelog.sh & - spustí jako samostatný proces skript měření a záznamu  
exit 0 - tento proces se ukončí
```

## 4.2 Skript měření a záznamu dat

Skript *writelog.sh* slouží k záznamu a ukládání dat. Jeho zjednodušená struktura je:

START

ověřit, zda-li existuje soubor *database.rrd*,

- pokud ne, vytvoří se nová databáze *rrdcreate database.rrd*

nekonečná smyčka *do...while* :

*./reset.sh*

*./m0 >> /dev/null* - změřit kartu 0., kanál 0. a výsledek zahodit, neboť karta 0 není obsazena

*ua=\$(./m0)* - změřit kartu 1., kanál 0 a výsledek uložit do proměnné *\$ua*

... - další kanály...

*uh=\$(./m0)* - změřit kartu 8., kanál 0 a výsledek uložit do proměnné *\$uh*

*./reset.sh*

*./m1 >> /dev/null* - změřit kartu 0., kanál 1. a výsledek zahodit, neboť karta 0 není obsazena

*ia=\$(./m1)* - změřit kartu 1., kanál 1 a výsledek uložit do proměnné *\$ia*

... - další kanály...

*ih=\$(./m1)* - změřit kartu 8., kanál 1 a výsledek uložit do proměnné *\$ih*

*t=\$(cat /sys/bus/w1/devices/10-0008020c07e6/w1\_slave | grep 't=' | cut -c 30-35)*

- načte do proměnné *\$t* upravenou hodnotu teploty

*echo `date +%s`, \$ua, \$ub, ... \$ig, \$ih, \$t >> cesta\_k\_souboru.csv*

-zapíše naměřené hodnoty na další řádek .csv souboru, který lze dále zpracovat na počítači v některém z tabulkových procesoru, jako například Microsoft Excel. Tento formát je čitelný přímo i pro lidskou obsluhu po otevření v textovém editoru, neboť se jedná o hodnoty oddělené čárkou, od čehož dostal tento formát i svůj název – Comma Separated Values.

*rrdtool graph soubor\_grafu.png ...* - vytvoří soubor obrázku grafu z databáze

*sleep 8* - čekání 8 sekund do dalšího měření

*done* – konec smyčky *do..while*, odtud se skript vrací na řádek s instrukcemi *do..while*

### 4.3 Získání hodnoty teploty

Ještě je třeba se zmínit o způsobu, jakým se získává údaj o teplotě:

Jestliže provedeme prostý výpis obsahu souboru příslušícího k 1-Wire čidlu, dostaneme podobný výpis:

```
root@minibian:~# cat /sys/bus/w1/devices/10-0008020c07e6/w1_slave
37 00 4b 46 ff ff 07 10 1e : crc=1e YES
37 00 4b 46 ff ff 07 10 1e t=27312
```

Jak je patrné, výpis obsahuje teplotu až na konci za řetězcem „t=“. Nechám si tedy vypsát řádek, který obsahuje tento řetězec:

```
root@minibian:~# cat /sys/bus/w1/devices/10-0008020c07e6/w1_slave | grep 't='
37 00 4b 46 ff ff 07 10 1e t=27312
```

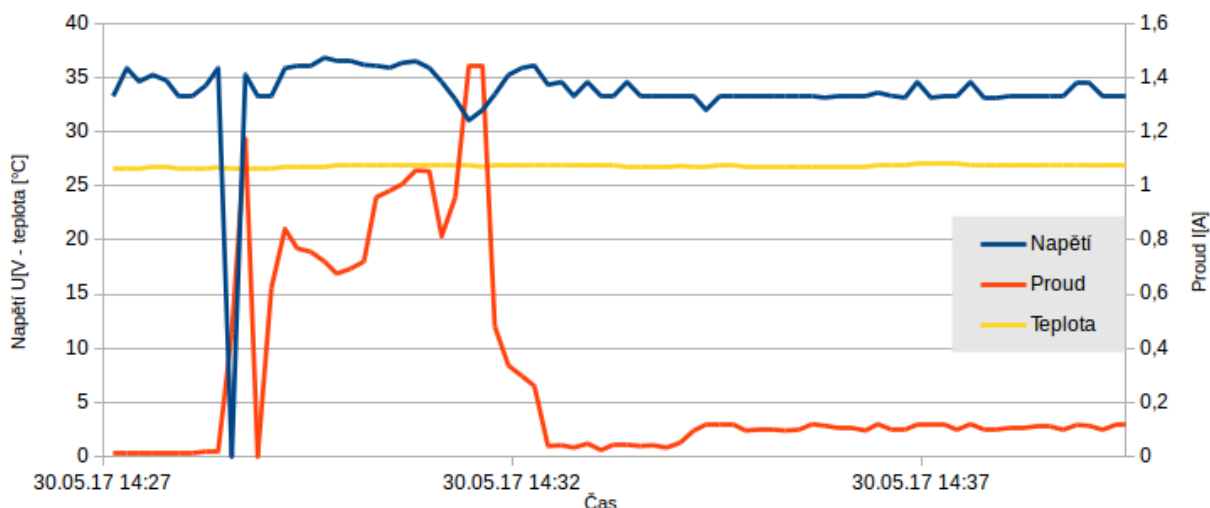
Oddělením posledních znaků dostávám samotnou hodnotu teploty v tisícinách °C

```
..@minibian:~# cat /sys/bus/w1/devices/10-0008020c07e6/w1_slave | grep 't=' | cut -c 30-35
27250
```

Takto získaná hodnota se přiřadí do proměnné *t*, která se pak dále zpracovává zápisem do .csv souboru a přidává skrze *rrdtool update* do databáze pro tvorbu grafu v reálném čase.

## 5. Naměřené údaje

První instalace zařízení ve Výukovém a a výzkumném centru v dopravě v Doubravících, neskončila úspěšně, neboť zařízení měřilo přibližně 20 minut nainstalování.

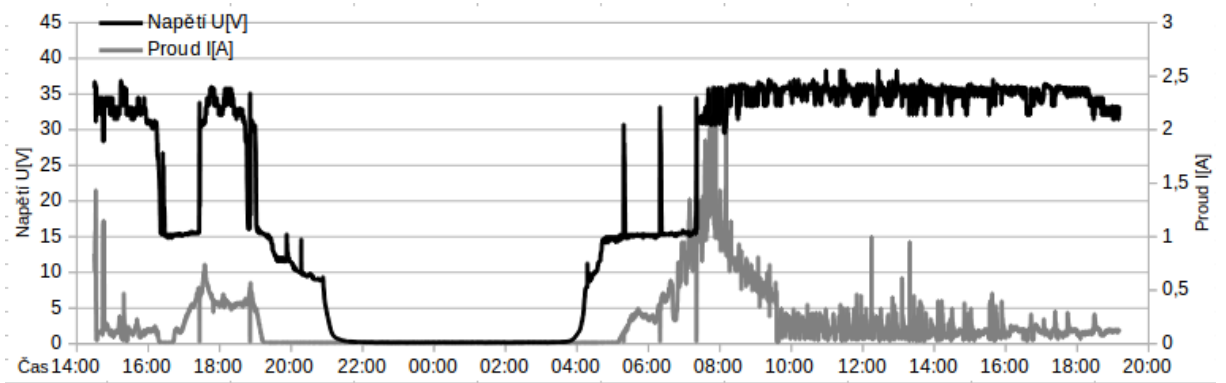


Obr. 13 Graf z prvního měření

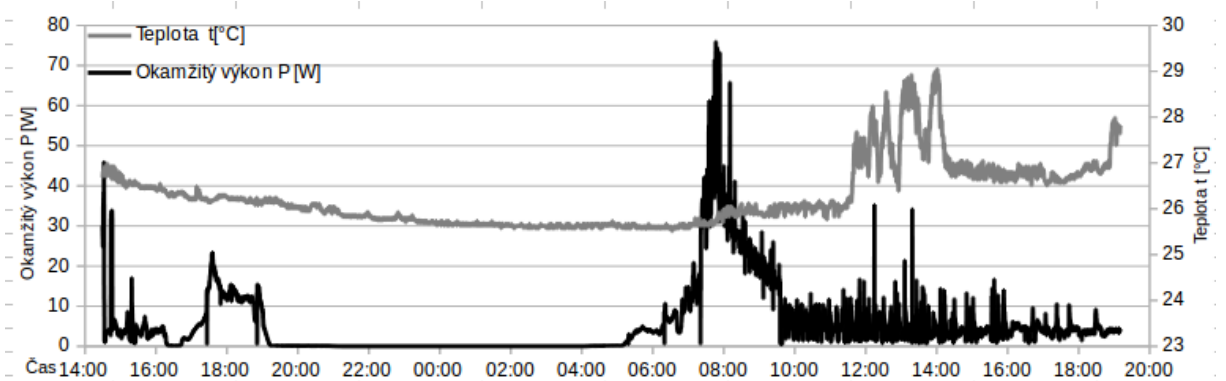
Příčinou byla patrně chyba ve startovacích skriptech, které jsem ještě na místě přizpůsobil a možná náhlý, byť krátkodobý, nepostřehnutelný výpadek elektrické energie, nebo jiný důvod způsobil, že se Raspberry Pi restartovalo a potom už měření a ukládání samovolně nezačalo.

Druhý den, jsem tedy vyzvedl přístroj bez naměřených dat. Příčinu problému jsem odstranil a jelikož bylo zapotřebí aby druhý pokus již vyšel, rozhodl jsem se zredukovat startovací skripty o záležitosti týkající se prodloužení životnosti karty. Došlo k přesunutí lokace naměřených dat z umístění dočasných souborů, odkud se pravidelně kopírovaly každou hodinu na kartu, do umístění přímo na kartě, aby zápis proběhl s jistotou. Negativní na této úpravě je, že se tím zvýšil počet zápisů na paměťovou kartu a tím z dlouhodobého hlediska klesla životnost paměťové karty. Další úpravou bylo vyřazení automatického přidělování IP adresy a nahrazení zápisem pro statickou adresu v souboru */etc/network/interfaces*

Druhé měření dopadlo o poznání lépe. Zaznamenal jsem čas od 14:29 do 19:10 druhého dne. Na cestě z Výzkumného a výukového centra v dopravě mě zastihl prudký déšť. Mraky zastínily slunce a způsobily náhlé snížení výstupního napětí solárních panelů. Jako zátěž byl u solárního panelu připojen nabíječ baterie. Na tuto baterii jsme připojili reostat, aby ráno došlo ke znatelnému dobíjení.



Obr. 14 Graf vytvořený z hodnot napětí a proudu solárního panelu



Obr. 15 Graf teploty a okamžitého výkonu (ten se zvlášť neměří, pro výpočet posloužily hodnoty napětí a proudu)

Výkyvy napětí v grafech jsou způsobeny momentálními světelnými podmínkami. Skokovitý přechod napětí není způsoben chybou v algoritmu převodu binární hodnoty na dekadickou. Možnou příčinou těchto skoků je to, že jsme měřili jeden z panelů, které byly v sérii, tudíž se mohly projevit jejich charakteristiky při částečném osvětlení, společně s přepínáním spínacího módu nabíječe. S tím patrně souvisí i občasné špičky.

Na základě připomínek jsem pátral po možné softwarové příčině těchto skoků, zjistil jsem, že špičky, které by poukazyvaly na softwarovou chybu nejsou pravidelné. Trvají také příliš dlouho, typicky okolo minuty a nejsou skokovitého charakteru.

## 6. ZÁVĚR

Cílem práce bylo navrhnout a zrealizovat automatické měřicí zařízení, pro bezobslužné měření a záznam žádaných veličin. Konkrétní požadavky na zařízení byly postupně v průběhu návrhu měněny a zpřesňovány s zařízením jsem těmto požadavkům přizpůsoboval. Výsledkem je nejen funkční zařízení, ale také snadno rozšiřitelná platforma pro další vývoj.

Do budoucna je možné funkce, které řídící mikropočítač Raspberry obsluhuje dále rozšířit. Z hlediska výpočetního výkonu je mou aplikací Raspberry ve verzi 3 využito jen velmi slabě a je zde rezerva pro další činnosti. Vzhledem k tomu, že grafický výstup je ve formě webové stránky, lze využít zařízení pro reprezentativní výstup například na informační LCD panel. Pro propojení s televizorem lze využít rozhraní HDMI, kterým je Raspberry Pi vybaveno. V tom případě se webový prohlížeč spustí přímo v Raspberry Pi.

Uvažovaným rozšířením je doplnění o RTC modul hodin reálného času například typu PCF85x3, nebo jiný. Tyto obvody se ve formě hotového modulu se zálohovací baterií připojují ke sběrnici I<sub>2</sub>C, která zůstala i pro tento důvod volná. Operační systém Linux podporuje velké množství těchto obvodů přímo v jádře a umí s nimi pracovat jako se standardním zdrojem času a není třeba nic dlouze programovat. Myslím, že to je jeden z příkladů, který velmi dobře popisuje princip vývoje jakékoliv aplikace v prostředí operačního systému GNU/Linux, neprogramovat nic nadvakrát a využívat jednoduché, ale účelné nástroje.

Z hlediska napájení se praktickým provozem zjistilo, že by bylo vhodné datalogger napájet z baterie, která je nainstalována u solárních panelů a je z nich dobíjena.

Nemám prozkoumány zdaleka všechny možnosti vývoje aplikací zahrnujících hardware a software, kterých je nepřeberné množství. Možnost tvořit vysoce funkční skripty jen pomocí nástrojů obsažených v systému, případně doinstalovaných, dle mého názoru skýtá pro mírně pokročilé uživatele možnost zprovoznit hardwarovou aplikaci s minimálními znalostmi o programování.

## 7. Seznam použité literatury

[1] BAREŠ, Miloš. *Elektrotechnické měření*. Pardubice: Univerzita Pardubice, 2006. ISBN 80-7194-910-8.

[2] VEDRAL, Josef a Jan FISCHER. *Elektronické obvody pro měřicí techniku*. Vyd. 2. Praha: Vydavatelství ČVUT, 2004. ISBN 80-01-02966-2.

[3] *Datasheet MCP3202* [online]. [cit. 2017-06-02]. Dostupné z: <http://www.farnell.com/datasheets/1669376.pdf>

[4] *Datasheet MCP602* [online]. [cit. 2017-06-02]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/21314g.pdf>

[5] *Datasheet MCP34063* [online]. [cit. 2017-06-02]. Dostupné z: <https://www.onsemi.com/pub/Collateral/MC34063A-D.PDF>

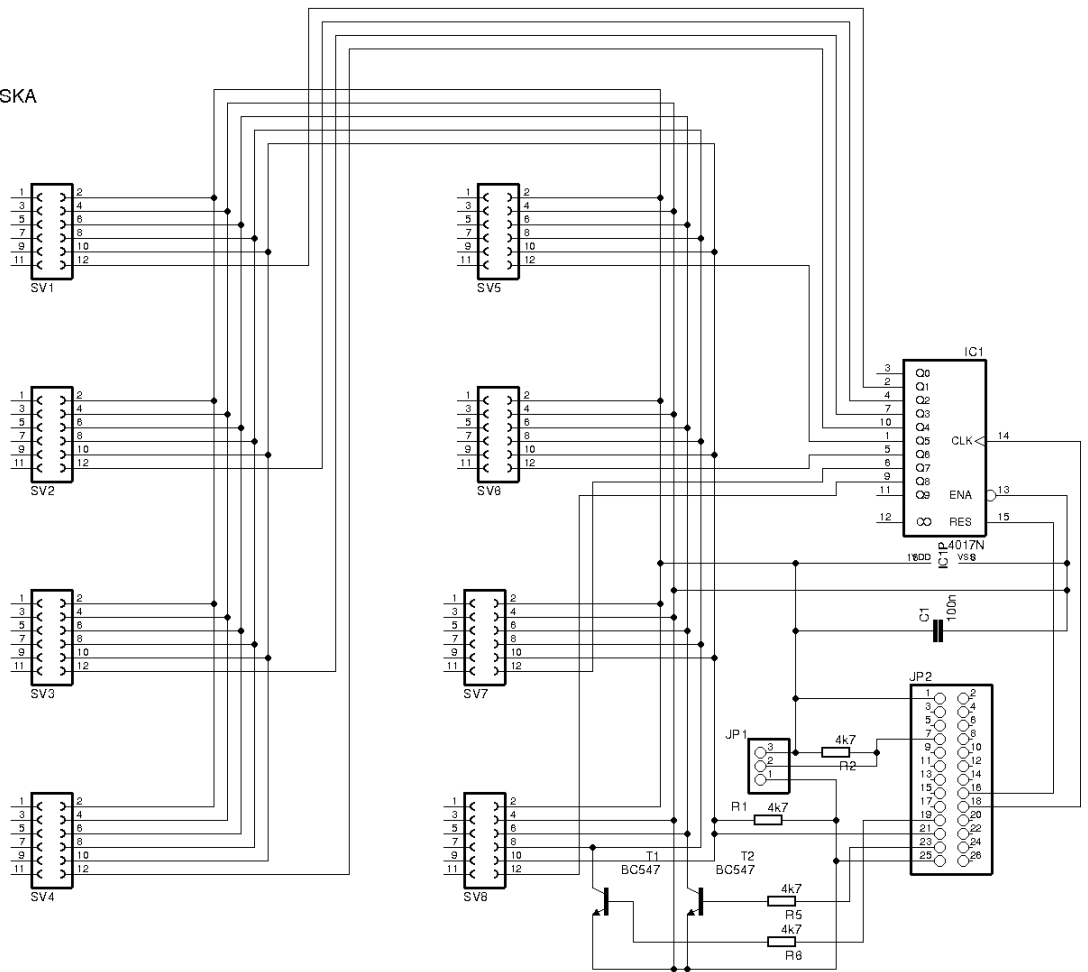
[6] Manuálové stránky k příkazům - man [online]. [cit. 2017-06-02]. Dostupné z: <https://linux.die.net/man/>

[7] Oficiální dokumentace k Raspberry Pi [online]. [cit. 2017-06-02]. Dostupné z: <https://www.raspberrypi.org/documentation/>

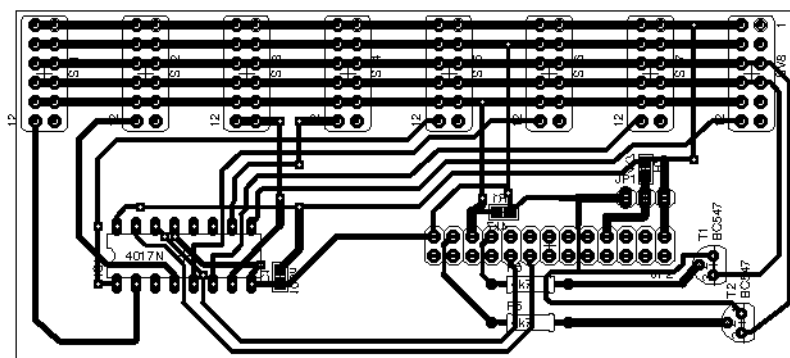


## 8. Přílohy

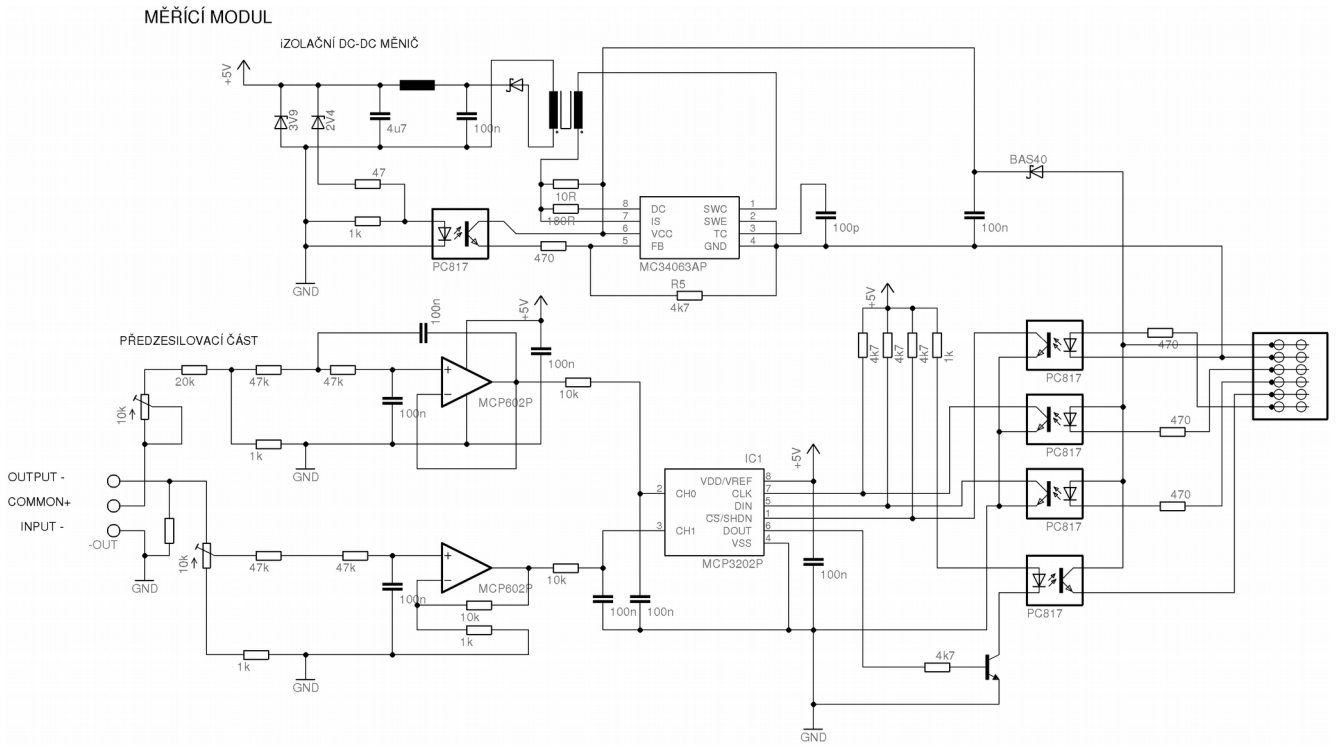
ZÁKLADNÍ DESKA



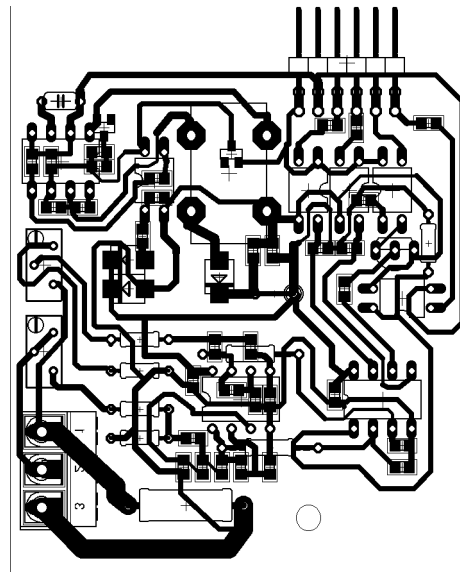
Obr. 16 Schéma zapojení základní desky



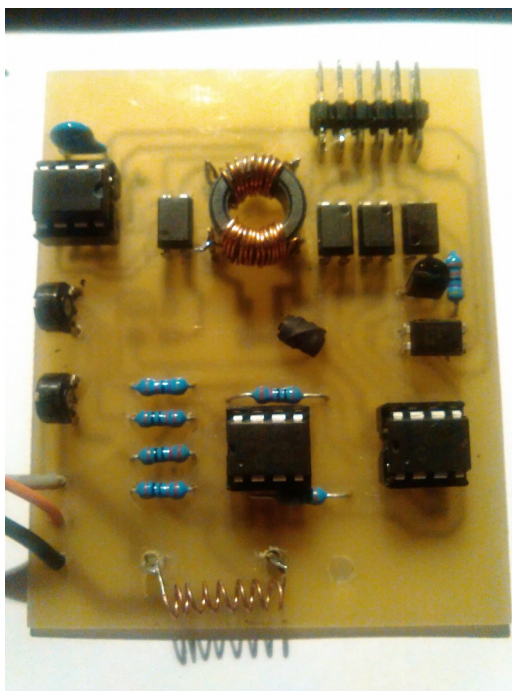
Obr. 17 Motiv desky s plošnými spoji - Základní deska



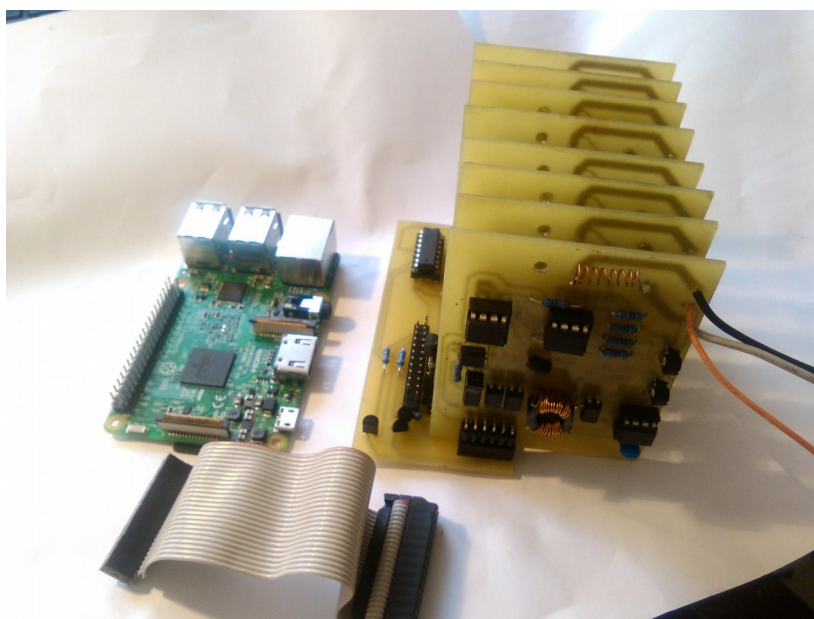
Obr. 18 Schéma zapojení měřicího modulu



Obr. 19 Motiv desky s plošnými spoji – Měřicí modul



*Obr. 20 - Osazený měřicí modul*



*Obr. 21 - Hotové sestavené zařízení, pro názornost odpojeno od Raspberry Pi*