

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

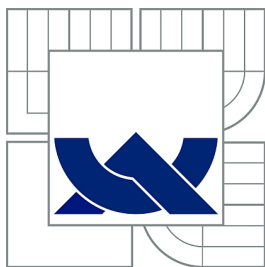
MĚŘENÍ SPOTŘEBY ELEKTRICKÉ ENERGIE  
VZDÁLENÝCH ZAŘÍZENÍ

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

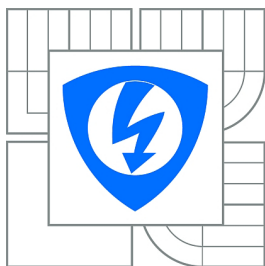
AUTOR PRÁCE  
AUTHOR

BC. LUKÁŠ MICHALÍK

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

## MĚŘENÍ SPOTŘEBY ELEKTRICKÉ ENERGIE VZDÁLENÝCH ZAŘÍZENÍ

MEASUREMENTS OF POWER CONSUMPTION ON REMOTE DEVICES

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

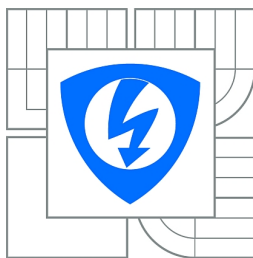
AUTOR PRÁCE  
AUTHOR

BC. LUKÁŠ MICHALÍK

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. ZBYNĚK FEDRA, PH.D.

BRNO 2013



**VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky  
a komunikačních technologií**

**Ústav radioelektroniky**

# Diplomová práce

magisterský navazující studijní obor  
**Elektronika a sdělovací technika**

**Student:** Bc. Lukáš Michalík  
**Ročník:** 2

**ID:** 115229  
**Akademický rok:** 2012/2013

## NÁZEV TÉMATU:

### **Měření spotřeby elektrické energie vzdálených zařízení**

## POKYNY PRO VYPRACOVÁNÍ:

Zhodnoťte možnosti měření spotřeby el.energie v komunikačních uzlech počítačové sítě. Navrhněte systém sběru dat z více měřených uzlů. Navrhněte způsob přenosu dat a platformu pro realizaci zařízení.

Navrhněte zapojení jednotlivých bloků zařízení, otestujte funkčnost dílčích částí a implementujte komunikační protokol pro jednotlivé části. Zhodnoťte rozsah přenášených dat, periodu sběru dat a případné lokální zálohování.

Oživte a otestujte navržené zařízení. Testování proveďte v existující síti. Prakticky ověřte funkčnost sběru dat. Navrhněte přehledný systém pro zobrazení a vyhodnocení získaných dat.

## DOPORUČENÁ LITERATURA:

[1] DONAHUE, G., A. Kompletní průvodce síťového experta. Vyd. 1. Brno: Computer Press, 2009, 528 s. ISBN 978-802-5122-471

[2] VÁGNER, M. Vzdálená správa jednočipových systémů. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 92 s. Vedoucí diplomové práce Ing. Pavel Kučera, Ph.D.

**Termín zadání:** 11.2.2013

**Termín odevzdání:** 24.5.2013

**Vedoucí práce:** Ing. Zbyněk Fedra, Ph.D.

**Konzultanti diplomové práce:**

**prof. Dr. Ing. Zbyněk Raida**  
*Předseda oborové rady*



## **ABSTRAKT**

Diplomová práce je zaměřena na digitální měření spotřeby elektrické energie a následný přenos naměřených výsledků vzdálenému k monitorovacímu centru po počítačové síti. Celá aplikace je navržena pro monitorování aktivních prvků počítačové sítě.

## **KLÍČOVÁ SLOVA**

spotřeba elektrické energie, výkon, mikrokontrolér, arm, vzdálená správa, SNMP, Raspberry Pi, Linux

## **ABSTRACT**

Thesis is focused on the digital measurement of power consumption and distribution the measured results to the remote monitoring center over computer network. The application is designed for monitoring of active elements computers network.

## **KEYWORDS**

power consumption, performance, microcontroller, arm, remote administration, SNMP, Raspberry Pi, Linux

MICHALÍK, L. Měření spotřeby elektrické energie vzdálených zařízení. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 97 s. Vedoucí diplomové práce Ing. Zbyněk Fedra, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Měření spotřeby elektrické energie vzdálených zařízení“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne .....

.....

(podpis autora)

## **Poděkování**

Děkuji Ing. Zbyňkovi Fedrovi, Ph.D., vedoucímu diplomové práce, za cenné rady a účinnou metodickou, pedagogickou a odbornou pomoc při zpracování diplomové práce.



# OBSAH

Úvod	14
<b>1 Způsob měření spotřeby elektrické energie</b>	<b>15</b>
1.1 Měření napětí . . . . .	15
1.2 Měření proudu . . . . .	15
1.3 Obvody pro měření výkonu . . . . .	16
<b>2 Systém sběru dat a jejich přenos</b>	<b>18</b>
2.1 Protokoly pro přenos dat, monitorování a řízení . . . . .	18
2.1.1 Popis protokolu SNMP . . . . .	18
2.2 Zpracování naměřených dat . . . . .	20
2.3 Monitorovací systémy . . . . .	21
2.3.1 Observium . . . . .	21
2.3.2 Zabbix . . . . .	22
2.3.3 Instalace Zabbix a Observium . . . . .	22
<b>3 Použité bloky</b>	<b>23</b>
3.1 Raspberry Pi . . . . .	23
3.1.1 Technická specifikace Raspberry Pi . . . . .	23
3.1.2 Operační systém . . . . .	24
3.1.3 Ovládání GPIO . . . . .	25
3.2 Měřicí modul . . . . .	27
3.2.1 Měřicí modul – první verze . . . . .	27
3.2.2 Obvod ADE7763 . . . . .	27
3.2.3 Optické oddělení . . . . .	28
3.2.4 DC   DC měnič . . . . .	28
3.2.5 Bistabilní relé . . . . .	28
3.2.6 Měřicí modul – druhá verze . . . . .	28
3.2.7 Náměty pro následující revizi . . . . .	29
<b>4 Implementace protokolů</b>	<b>31</b>
4.1 Implementace protokolu SNMP . . . . .	31
4.1.1 Instalace SNMP . . . . .	32
4.1.2 Konfigurace SNMPv2c . . . . .	32
4.1.3 Konfigurace SNMPv3 . . . . .	34
4.1.4 Další možnosti nastavení SNMP . . . . .	36
4.2 Implementace ostatních protokolů . . . . .	38
4.2.1 Protokol SSH . . . . .	38

4.2.2	Protokol HTTP . . . . .	39
<b>5</b>	<b>Přenášená data, perioda sběru dat a lokální zálohování</b>	<b>40</b>
5.1	Velikost přenášených dat . . . . .	40
5.2	Zatížení klienta a agenta při čtení dat . . . . .	41
5.3	Perioda sběru dat . . . . .	42
5.4	Lokální zálohování . . . . .	42
5.4.1	RAM disk . . . . .	43
<b>6</b>	<b>Měření</b>	<b>44</b>
6.1	Měřicí Software . . . . .	44
6.2	Spouštění daemon-a . . . . .	47
6.3	Sestavení měření . . . . .	48
6.3.1	Raspberry Pi – podmínky měření . . . . .	48
6.3.2	Systém UBiQUiTi mFi – podmínky měření . . . . .	49
6.3.3	METEX – podmínky měření . . . . .	49
6.4	Měření – výsledky . . . . .	50
6.5	Porovnání měření Raspberry Pi, UBiQUiTi mFi a METEX . . . . .	51
6.5.1	Měření frekvence . . . . .	51
6.5.2	Měření napětí . . . . .	52
6.5.3	Měření výkonu . . . . .	52
6.5.4	Chyby měření . . . . .	52
6.6	Vzdálené měření . . . . .	53
<b>7</b>	<b>Porovnání vlastností systému Raspberry Pi se systémem UBiQUiTi mFi</b>	<b>63</b>
<b>8</b>	<b>Závěr</b>	<b>64</b>
	<b>Literatura</b>	<b>66</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>69</b>
	<b>Seznam příloh</b>	<b>70</b>
<b>A</b>	<b>Monitorování několika nezávislých míst</b>	<b>71</b>
<b>B</b>	<b>Zapojení Síťových prvků</b>	<b>72</b>
<b>C</b>	<b>Měření Zabbix</b>	<b>73</b>
<b>D</b>	<b>Schémata – verze první</b>	<b>75</b>

E	Desky plošných spojů – verze první	78
F	Náhled DPS – verze první	80
G	Schémata – verze druhá	81
H	Desky plošných spojů – verze druhá	85
I	Náhled DPS – verze druhá	89
J	Seznam součástek – verze první	90
K	Seznam součástek – verze druhá	91
L	Konfigurační soubory	93

# SEZNAM OBRÁZKŮ

1.1	Digitální voltmetr s obvodem ICL7107 . . . . .	15
2.1	Organizace SNMP MIB se zvýrazněnou cestou k systémovým informacím . . . . .	20
4.1	Náhled na webovou stránku umístěnou na webovém serveru Apache spuštěném přímo na Raspberry Pi . . . . .	39
5.1	Zatížení monitorovacího serveru při čtení SNMP dat . . . . .	41
5.2	Zatížení monitorovacího serveru při cyklickém spouštění příkazu snmpwalk . . . . .	41
5.3	Zatížení Raspberry Pi měřicím programem . . . . .	42
6.1	Hlavní struktura kódu . . . . .	46
6.2	Průběh odebíraného proudu za 24 hod. – měřeno pomocí Raspberry Pi	54
6.3	Průběh odebíraného proudu za 24 hod. – měřeno pomocí METEX . .	54
6.4	Průběh odebíraného proudu za 24 hod. – měřeno pomocí UBiQUiTi mFi . . . . .	54
6.5	Průběh odebíraného proudu zoom 15:00 – 15:35 hod. – měřeno pomocí UBiQUiTi mFi . . . . .	55
6.6	Průběh odebíraného proudu zoom 15:00 – 15:35 hod. – měřeno pomocí Raspberry Pi . . . . .	55
6.7	Průběh odebíraného proudu zoom 15:00 – 15:35 hod. – měřeno pomocí METEX . . . . .	56
6.8	Průběh odebíraného proudu zoom 17:01 – 17:41 hod. – měřeno pomocí UBiQUiTi mFi . . . . .	57
6.9	Průběh odebíraného proudu zoom 17:01 – 17:41 hod. – měřeno pomocí Raspberry Pi . . . . .	57
6.10	Průběh odebíraného proudu zoom 17:01 – 17:41 hod. – měřeno pomocí METEX . . . . .	58
6.11	Průběh odebíraného proudu zoom 11:52 – 12:02 hod. – měřeno pomocí UBiQUiTi mFi . . . . .	59
6.12	Průběh odebíraného proudu zoom 11:52 – 12:02 hod. – měřeno pomocí Raspberry Pi . . . . .	59
6.13	Průběh odebíraného proudu zoom 11:52 – 12:02 hod. – měřeno pomocí METEX . . . . .	60
6.14	Průběh síťové frekvence za 24 hod. – měřeno pomocí Raspberry Pi . .	61
6.15	Průběh síťového napětí za 24 hod. – měřeno pomocí Raspberry Pi . .	61
6.16	Průběh odebíraného výkonu za 24 hod. – měřeno pomocí UBiQUiTi mFi . . . . .	61

6.17	Absolutní odchylka měření Raspberry Pi od měření přístrojem METEX, v porovnání s měřenou hodnotou proudu přístrojem METEX . . . . .	62
6.18	Absolutní odchylka měření Raspberry Pi od měření přístrojem METEX (amplituda vynásobena 5), v porovnání s měřenou hodnotou proudu přístrojem METEX . . . . .	62
C.1	Průběh odebíraného napětí – měřeno systémem Zabbix . . . . .	73
C.2	Průběh odebíraného proudu – měřeno systémem Zabbix . . . . .	73
C.3	Průběh odebíraného výkonu – měřeno systémem Zabbix . . . . .	74
C.4	Průběh síťové frekvence – měřeno systémem Zabbix . . . . .	74
D.1	Schéma zapojení napájecího obvodu – (Tranformless) . . . . .	75
D.2	Schéma zapojení modulu pro Raspberry Pi – verze první . . . . .	76
D.3	Schéma zapojení obvodu ADE7763 – verze první . . . . .	77
E.1	Deska plošných spojů – Top – verze první . . . . .	78
E.2	Deska plošných spojů – Bottom – verze první . . . . .	78
E.3	Deska plošných spojů – osazovací předloha – verze první . . . . .	79
E.4	Deska plošných spojů – 3D náhled – verze první . . . . .	79
F.1	Náhled na první verzi měřicího modulu určeného pro Raspberry Pi – po dodatečných úpravách . . . . .	80
G.1	Schéma zapojení obvodu ADE7763 – verze druhá . . . . .	81
G.2	Schéma zapojení optického oddělení a relé – verze druhá . . . . .	82
G.3	Schéma zapojení přepěťové ochrany – verze druhá . . . . .	83
G.4	Schéma zapojení měřicích bodů a LED diod – verze druhá . . . . .	84
H.1	Deska plošných spojů – Top – verze druhá (zvětšení 0,65) – návrh obsahuje i rozšiřující modul pro LCD displej, ten však není v této práci využit . . . . .	85
H.2	Deska plošných spojů – Bottom – verze druhá (zvětšení 0,75) – návrh obsahuje i rozšiřující modul pro LCD displej, ten však není v této práci využit . . . . .	86
H.3	Deska plošných spojů – osazovací předloha – verze druhá (zvětšení 0,75) – návrh obsahuje i rozšiřující modul pro LCD displej, ten však není v této práci využit . . . . .	87
H.4	Deska plošných spojů – 3D náhled – verze druhá (zvětšení 0,75) – návrh obsahuje i rozšiřující modul pro LCD displej, ten však není v této práci využit . . . . .	88
I.1	Náhled na druhou verzi měřicího modulu určeného pro Raspberry Pi . . . . .	89

# SEZNAM TABULEK

1.1	Přehled obvodů firmy ANALOG DEVICES určených pro měření výkonu	17
2.1	Popis hlavních podvětví SNMP MIB . . . . .	20
3.1	Technické parametry Raspberry Pi (Model B revize 2) . . . . .	23
6.1	Přehled událostí při celodenním měření odběru . . . . .	50
6.2	Chyby měření METEX . . . . .	52
J.1	Seznam použitých součástek – první verze měřicího modulu . . . . .	90
K.1	Seznam použitých součástek – druhá verze měřicího modulu . . . . .	91

# ÚVOD

Snahou každého člověka, popřípadě firmy je minimalizovat náklady na provoz elektrických zařízení a co nejvíce tak ušetřit. Zvýšená spotřeba zařízení neznamena jen větší finanční náklady. Mnohdy se může jednat o závadu zařízení nebo také špatnou konfiguraci daného zařízení. V případě jediného sledovaného zařízení to obsluhu nepřináší neřešitelný problém. Potíže nastanou, je-li sledováno více topologicky vzdálených zařízení. Takovéto prostředí je známé například z počítačových sítí, kde jsou desítky síťových prvků rozmístěných například na ploše jedné budovy, města nebo státu.

Cílem diplomové práce „Měření spotřeby elektrické energie vzdálených zařízení“ je navrhnout systém pro měření spotřeby elektrické energie a přenos naměřených dat po počítačové síti a navrhnout zařízení se standardizovaným rozhraním pro předávání dat. Samotným zobrazováním a zpracováváním dat se tato práce bude zabývat pouze okrajově, formou doporučení.

Hlavní myšlenkou, kterou se práce zabývá, je návrh kompaktního zařízení, které bude snadno instalovatelné a použitelné pro měření jakéhokoliv typu elektrického spotřebiče. Uživateli bude poskytovat naprostou volnost při práci s naměřenými údaji a zároveň ho bude jednoduchou a univerzální formou informovat o současném stavu zařízení. Samotný text rozebírá možnosti měření spotřeby elektrické energie. Podrobně popisuje komunikační protokol SNMP, pomocí kterého lze přenášet veškeré informace o měření po počítačové síti. Práce řeší kapacitní zatížení počítačové sítě přenášenými daty a vytížení jednotlivých prvků při odečtu hodnot. Popisuje navržené zařízení a pojednává o jeho přednostech a nedostatcích zaznamenaných v rámci testování.

V závěru práce jsou vyhodnocena naměřená data a navržené zařízení je srovnáno s komerčně dostupným produktem podobného charakteru, pracujícím však na jiném principu.

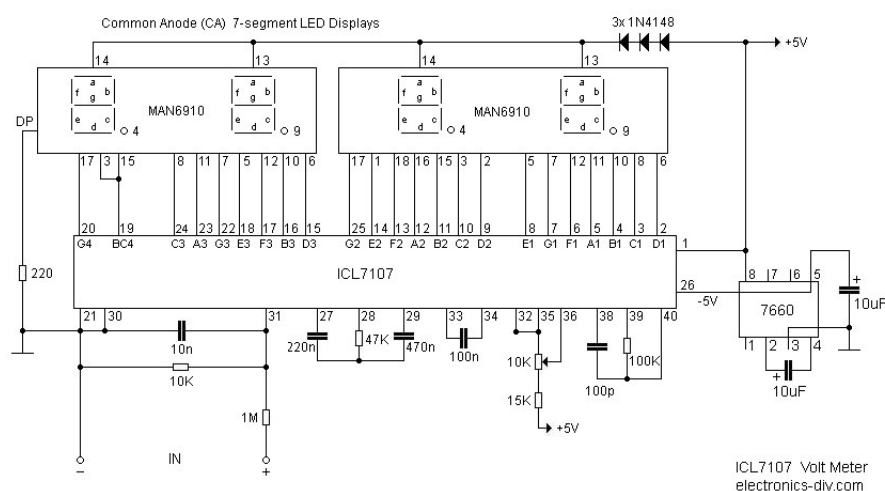
# 1 ZPŮSOB MĚŘENÍ SPOTŘEBY ELEKTRICKÉ ENERGIE

V této kapitole budou stručně popsány možnosti měření spotřeby elektrické energie. V závěru bude vybrána jedna varianta, která bude v rámci projektu dále využívána.

Spotřeba elektrické energie a odebíraný výkon se nedá měřit přímo. Ve většině případů se tyto údaje odvozují z napájecího napětí a odebíraného proudu.

## 1.1 Měření napětí

Pomineme-li analogové možnosti měření napětí, jedná se v dnešní době většinou o zapojení digitálních jednoúčelových obvodů, které jsou určeny pro měření napětí a jsou většinou vybaveny i výstupem pro LCD displej nebo sedmisegmentový displej. Příklad takového zapojení s obvodem ICL7107 můžeme vidět na obrázku č.1.1.



Obr. 1.1: Digitální voltmetr s obvodem ICL7107 [10]

Dalším, dnes často aplikovaným řešením je využití mikrokontroléru, respektive jeho A/D převodníků. Měřené napětí se přes „kalibrovaný“ odporový dělič přivede na A/D převodník. Mikrokontrolér jej převede do digitální podoby, přepočítá na skutečnou hodnotu a zobrazí na displeji.

## 1.2 Měření proudu

K měření proudu se dá přistupovat dvěma základními metodami. Jednou variantou je použití bočníku a následné měření napětí na bočníku. Zde se postupuje obdobně,



jak bylo uvedeno v předchozím odstavci o měření napětí. Nastává tady však problém s velikostí bočníku. Při měření velkých proudů je zapotřebí bočníku s velice malým odporem, což komplikuje situaci při měření malých proudů, kdy úbytek napětí na bočníku je minimální. Měřené hodnoty potom zkresluje i šum.

Druhou variantou je použití indukčních metod. Avšak po prozkoumání katalogových listů současných klešťových ampérmetrů lze konstatovat jejich nevyhovující přesnost (pro větší měřicí rozsahy), a to i u velmi drahých zařízení. Jedná se o 10 – 100 mA. Požadavky na možnosti, přesnost a rozsah měřených veličin tohoto projektu jsou vyšší.

### 1.3 Obvody pro měření výkonu

Perspektivní se jeví využití specifických obvodů, které kombinují měření napětí s měřením proudu (bočnickovou metodou). Tyto obvody mají v proudové větvi velice citlivé diferenční zesilovače. Lze tedy při standardním zapojení s bočníkem  $0.01\ \Omega$  měřit na rozsahu 2 A [1]. (Zařízení od přibližně 50 W.) Při potřebě měřit zařízení s menší spotřebou elektrické energie, bude potřeba použít bočníku s větší ohmickou hodnotou.

V oblasti měření spotřeby elektrické energie jednofázově napájených zařízení jsou v současné době dostupné obvody několika firem, například STMicroelectronics [16], Microchip Technology Inc.[8], SAMES [19] a ANALOG DEVICES [21].

Pro zpracování této práce byly vybrány obvody firmy ANALOG DEVICES, proto se jimi bude následující text zabývat podrobněji. Zejména pak obvodem ADE 7763. Obvod je schopen měřit všechny elektrické veličiny (napětí, proud a výkon), disponuje komunikační sběrnici SPI a je zapouzdřen v pouzdře lehce zpracovatelném bez speciálního vybavení. V tabulce č.1.1 jsou uvedeny jednotlivé obvody od této firmy (včetně jejich vlastností) dostupné při psaní této práce.

**Tab. 1.1:** Přehled obvodů firmy ANALOG DEVICES určených pro měření výkonu [21]

Single Phase Metering		Measurement Parameters		Current Sensor Interface		On-Chip MCU	Anti-Tamper Support	Output Interface	Package
Part#	Query Parameter			Current Sensor Interface		On-Chip MCU	Anti-Tamper Support	Output Interface	Package
ADE7880	VA, VAR, Vrms, Watt			Current Transformer, Rogowski Coil				Pulsed	40-Lead LFQSP
ADE7953	Irms, VA, VAR, Vrms, Watt			Current Transformer, Rogowski Coil			Current Fault Missing Neutral	HSDC, I2C, Pulsed, SPI	28-Lead LFQSP
ADE7854	Irms, VA, Vrms, Watt			Current Transformer, Rogowski Coil				HSDC, I2C, Pulsed, SPI	40-Lead LFQSP
ADE7858	Irms, VA, VAR, Vrms, Watt			Current Transformer, Rogowski Coil				HSDC, I2C, Pulsed, SPI	40-Lead LFQSP
ADE7868	Irms, VA, VAR, Vrms, Watt			Current Transformer, Rogowski Coil			Current Fault	HSDC, I2C, Pulsed, SPI	40-Lead LFQSP
ADE7878	Irms, VA, VAR, Vrms, Watt			Current Transformer, Rogowski Coil			Current Fault	HSDC, I2C, Pulsed, SPI	40-Lead LFQSP
ADE5569	Irms, VA, Vrms, Watt			Current Transformer, Rogowski Coil, Shunt		8052	Missing Neutral	I2C, Pulsed, SPI, UART	64-Lead LQFP
ADE7518	Irms, VA, VAR, Vrms, Watt			Current Transformer, Shunt		8052		I2C, Pulsed, SPI, UART	64-Lead LQFP
ADE5169	Irms, VA, VAR, Vrms, Watt			Current Transformer, Rogowski Coil, Shunt		8052	Current Fault, Missing Neutral	I2C, Pulsed, SPI, UART	64-Lead LQFP
ADE7116	Irms, VA, Vrms, Watt			Current Transformer, Shunt		8052	Current Fault, Missing Neutral	I2C, Pulsed, SPI, UART	64-Lead LQFP
ADE7156	Irms, VA, Vrms, Watt			Current Transformer, Shunt		8052	Current Fault, Missing Neutral	I2C, Pulsed, SPI, UART	64-Lead LQFP
ADE7166	Irms, VA, Vrms, Watt			Current Transformer, Shunt		8052	Current Fault, Missing Neutral	I2C, Pulsed, SPI, UART	64-Lead LQFP
ADE7169	Irms, VA, VAR, Vrms, Watt			Current Transformer, Rogowski Coil, Shunt		8052	Current Fault, Missing Neutral	I2C, Pulsed, SPI, UART	64-Lead LQFP
ADE7569	Irms, VA, VAR, Vrms, Watt			Current Transformer, Rogowski Coil, Shunt		8052	Current Fault, Missing Neutral	I2C, Pulsed, SPI, UART	64-Lead LQFP
ADE7566	Irms, VA, Vrms, Watt			Current Transformer, Shunt		8052	Missing Neutral	I2C, Pulsed, SPI, UART	64-Lead LQFP
ADE7761B	Watt			Current Transformer, Shunt		8052	Missing Neutral	I2C, Pulsed, SPI, UART	64-Lead LQFP
ADE7762	Watt			Current Transformer			Current Fault, Missing Neutral	Pulsed	20-Lead SSOP
ADE7757A	Watt			Current Transformer, Shunt				Pulsed	24-Lead SOIC
ADE7752A	Watt			Current Transformer				Pulsed	16-Lead SOIC
ADE7768	Watt			Current Transformer, Shunt				Pulsed	24-Lead SOIC
ADE7769	Watt			Current Transformer, Shunt				Pulsed	16-Lead SOIC
ADE7763	Irms, VA, Vrms, Watt			Current Transformer, Rogowski Coil, Shunt				Pulsed, SPI	20-Lead SSOP
ADE7758	Irms, VA, VAR, Vrms, Watt			Current Transformer, Rogowski Coil				Pulsed, SPI	24-Lead SOIC
ADE7753	Irms, VA, VAR, Vrms, Watt			Current Transformer, Rogowski Coil, Shunt				Pulsed, SPI	20-Lead SSOP
ADE7754	Irms, VA, Vrms, Watt			Current Transformer, Shunt				Pulsed, SPI	24-Lead SOIC
ADE7751	Watt			Current Transformer, Shunt			Current Fault	Pulsed	24-Lead SSOP
ADE7759	Watt			Current Transformer, Rogowski Coil, Shunt				Pulsed, SPI	20-Lead SSOP
ADE7755	Watt			Current Transformer				Pulsed	24-Lead SSOP

## 2 SYSTÉM SBĚRU DAT A JEJICH PŘENOS

Jak bylo v úvodu uvedeno, zařízení bude provádět měření spotřeby na různých místech rozlehlější počítačové sítě, obr. A. V této kapitole bude rozebráno, jaké protokoly bude zařízení používat pro komunikaci a distribuci dat, jak bude do sítě připojeno a jakým způsobem bude nakládáno s naměřenými daty.

### 2.1 Protokoly pro přenos dat, monitorování a řízení

Je celá řada možností (protokolů), pomocí kterých se dá v dnešní době komunikovat se vzdáleným zařízením. Jednou z nich je protokol SNMP popsáný v dokumentu RFC 1157 [11]. Jedná se o velice starý protokol pocházející z roku 1990. V dnešní době existuje již třetí verze tohoto protokolu. Přičemž verze druhá rozšiřuje první verzi o autentizaci a verze třetí o šifrovaný přenos. SNMP protokol byl vyvinut pro potřeby správy sítě. Proto nám poskytuje dobré prostředky pro sběr dat z různých míst sítě, jejich následné zpracování a monitorování.

#### 2.1.1 Popis protokolu SNMP [12]

Jedná se o asynchronní protokol založený na bázi klient/server. Až na výjimky komunikaci zahajuje SNMP klient dotazem na SNMP agenta (SNMP server), který na dotazy odpovídá. Ve výjimečných situacích může SNMP agent vyslat tzv. trapy. Jedná se o UDP pakety směřované ke klientovi. Trapy jsou využívány pouze v případě odeslání informace o předem definovaných chybových stavech. Běžná komunikace se provádí přes port 161. Případný trap je odeslán na port 162. Pro účely této práce bude na zařízení spuštěn SNMP agent, který bude poskytovat informace o aktuální spotřebě monitorovacím serverům, které se budou připojovat prostřednictvím SNMP klienta.

SNMP protokol používá ve verzích 1 a 2 následující příkazy (převzato z: [12]):

- **get-request** – získání informace z MIB,
- **get-next-request** – umožňuje klientovi získat informace o objektech v MIB bez znalosti jejich přesných jmen, umožňuje postupné procházení celým hierarchickým stromem,
- **set-request** – změna hodnoty proměnné agenta,

- **trap** – jediný typ příkazu vysílaný bez předchozího vyžádání, agent jej zasílá klientovi jako reakci na specifikovanou událost, zpráva zůstává nepotvrzená, proto agent nemá jistotu, zda byla doručena,
- **get-response** – agent vykoná tuto operaci jako reakci na předchozí příkazy – je to vlastně odpověď agenta klientovi. Odpověď obsahuje i dotaz, protože protokol nezajišťuje souvislost mezi dotazem a odpovědí,
- **get-bulk** – operace, která je součástí SNMP v. 2. Umožňuje vyžádat si k přečtení celou skupinu informací z MIB, čímž se mnohdy urychluje komunikace,
- **inform** – umožňuje komunikaci dvou klientů mezi sebou.

## Databáze MIB

MIB je databáze stromové struktury. Aby byla možná komunikace mezi klientem a serverem, je zapotřebí, aby tuto strukturu klient znal. Struktura dat je orientována objektově. Jednotlivá data se ukládají do objektů a jsou sdružována do tříd. Jednotlivé objekty mohou obsahovat další objekty nebo jiné třídy. Každý agent by měl obsahovat základní strukturu MIB databáze. Přidávání vlastních informací se provádí definicí nové MIB, přidáním experimentální větve, nebo přidáním objektů v rámci podstromu soukromé větve. V rámci přesné specifikace se z databáze MIB vylučují záznamy, jejichž hodnotu lze získat z jiných záznamů v databázi již uvedených, například kombinací několika záznamů.

Pro ilustraci a lepší pochopení problematiky je na stránkách ipMonitor [23] k dispozici interaktivní strom MIB s možností vyhledávání.

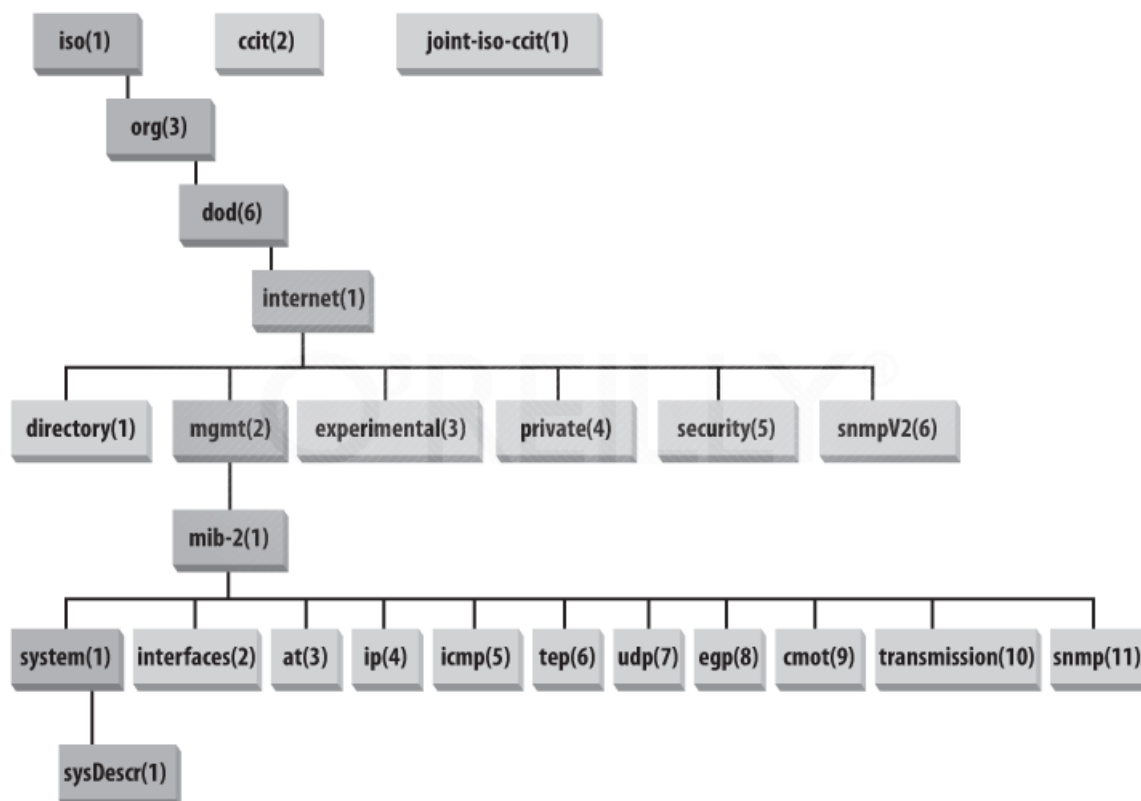
Obrázek č. 2.1 [15] zobrazuje základní strukturu SNMP MIB stromu. Na obrázku je zvýrazněna cesta k systémovým informacím. K těmto informacím lze přistoupit pomocí žádosti klienta, jejíž ukázka (využití příkazu `snmpwalk`) je uvedena v příkladu výpis: 2.1. Podrobnější popis této problematiky je uveden dále v textu.

V tabulce č. 2.1 je stručný popis podvětví SNMP MIB.

**Výpis: 2.1:** Příklad přístupu k systémovým informacím zobrazeným na obrázku č. 2.1 [15]

```
snmpwalk -v 1 -c public IPaddress .1.3.6.1.2.1.1.1.0

snmpwalk -v 1 -c public IPaddress .iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0
```



**Obr. 2.1:** Organizace SNMP MIB se zvýrazněnou cestou k systémovým informacím [15]

**Tab. 2.1:** Popis hlavních podvětví SNMP MIB [15]

Podstrom	Popis
directory(1)	OSI directory
mgmt(2)	RFC standard objects
experimental(3)	Internet experiments
private(4)	Vendor-specific
security(5)	Security
snmpV2(6)	SNMP internals

## 2.2 Zpracování naměřených dat

Naměřená data aktuálního výkonu se budou dále distribuovat pomocí SNMP protokolu, jak bylo uvedeno výše. Z toho plynou následující předpoklady, jak se bude s daty nakládat.

## **Distribuce dat**

Data budou odesílána striktně pouze na vyžádání monitorovacího serveru, popřípadě monitorovacích serverů.

## **Uchovávání dat**

Data budou uchovávána pouze na nezbytně nutnou dobu. Zálohování a zpracování dat bude náležet monitorovacím serverům.

## **Sběrná místa – Monitorovací servery**

Zařízení bude pracovat samostatně, bez závislosti na počtu a stavu monitorovacích serverů.

## **Zpracování dat v samotném zařízení**

V případě požadavku na zpracování dat přímo v zařízení bude tento požadavek realizován nasazením SNMP klienta přímo do zařízení. Rozhraní samotného agenta se tak nezmění.

# **2.3 Monitorovací systémy**

Existuje celá řada monitorovacích serverů a systémů. V následující kapitole jsou zmíněny dva, které byly v rámci práce testovány. Jedná se o systémy Observium [13] a Zabbix [26].

## **2.3.1 Observium**

Observium [13] je relativně mladý monitorovací systém. První záznamy o něm se objevují kolem roku 2007 a je distribuován pod licencí QPL-1.0. Na první pohled se systém jeví jako dynamický a efektní. Je velice jednoduchý na konfiguraci a uživatelsky přívětivý. Poskytuje celou řadu grafů a možností zobrazení naměřených dat. Po instalaci systému na monitorovací server je jednoduše přidáno zařízení, které je pomocí protokolu SNMP prohledáno a následně zařazeno k trvalému sledování. Pro přidání dalšího monitorovaného zařízení stačí zadat jeho doménové jméno a informace o SNMP verzi, popřípadě přihlašovací údaje.

Tím však výhody toho systému končí. V praxi ne všechna zařízení vlastní individuální doménové jméno. Někdy je potřeba přistoupit na zařízení přes jeho IP adresu (sítě menšího rozsahu). Tento problém se dá však jednoduše řešit přidáním záznamu do `/etc/hosts` v OS, na kterém Observium běží.

Observium podporuje celou řadu komerčně dostupných zařízení. Tato vlastnost mu přidává na kvalitě, ale také značně omezuje pole působnosti. Pro jednotlivá zařízení jsou definovány konkrétní větve v MIB. Observium při prohledávání MIB zkoumá pouze známé části tohoto stromu, což vylučuje použití uživatelské větve. Je možné nastavit SNMP agenta běžícího na monitorovaném zařízení tak, aby ukládal data do částí MIB, které Observium prohledává, ale tato varianta je značně nepraktická.

### **2.3.2 Zabbix**

Zabbix [26] je o něco starší systém než Observium. Vývoj probíhá od roku 2001. Systém je distribuován pod licencí GNU General Public License (GPL) version 2. Jedná se také o volně dostupný systém. Instalace tohoto systému vyžaduje lepší administrátorské znalosti než tomu je u Observia. Konfigurace samotného systému je komplexnější, a tím i složitější. Zabbix dává uživateli, respektive správci větší volnost v nastavení, a z toho vyplývá i větší variabilita celého systému. Díky jeho vlastnostem mohlo být otestováno monitorování hodnot, naměřených pomocí Raspberry Pi, systémem Zabbix.

### **2.3.3 Instalace Zabbix a Observium**

Oba systémy jsou velice dobře zdokumentovány na domovských stránkách [26] a [13]. Proto by byl popis instalace a nastavení těchto systémů v tomto dokumentu bezpředmětný a s postupem času snad až zavádějící.

## 3 POUŽITÉ BLOKY

Na vyvíjené zařízení jsou kladeny nároky především na malou spotřebu elektrické energie a na dobré vlastnosti týkající se síťové komunikace. Tyto parametry splňuje komerčně dostupný „mini počítač“ Raspberry Pi. Jeho výhody tkví především ve velice malé spotřebě elektrické energie, cenové dostupnosti a především v komplexním řešení, které je velice snadno rozšiřitelné do jakékoliv aplikace.

### 3.1 Raspberry Pi

Pro řídicí a komunikační část zařízení byla vybrána deska Raspberry Pi. V následujících kapitolách jsou rozebrány technické parametry a softwarové požadavky, respektive možnosti.

#### 3.1.1 Technická specifikace Raspberry Pi

Jedná se o  $55 \times 85$  mm velkou desku s parametry: tabulka 3.1 [3], [6] a [7].

**Tab. 3.1:** Technické parametry Raspberry Pi (Model B revize 2)

**Osazeno:**

Processor:	ARM1176JZF-S 700 MHz ARM11 family
Memory:	(SDRAM) 512 Megabytes (MiB)
Čipset:	Broadcom BCM2835 (CPU, GPU, DSP, and SDRAM)
Grafický čip:	Broadcom VideoCore IV, OpenGL ES 2.0, 1080p30 h.264/MPEG-4 AVC high-profile decoder

**Rozhraní:**

Video výstup:	Composite RCA, HDMI
Audio výstup:	3.5 mm jack, HDMI
Paměťové médium:	SD, MMC, SDIO card slot
Ethernet:	10/100 RJ45
USB:	$2 \times 2.0$
Display:	Slot na připojení displeje
Kamera:	Slot na připojení kamery
GPIO:	11 vstupně výstupních pinů včetně hardwarové podpory SPI, I <sup>2</sup> C, serial UART
Programátorské rozhraní:	JTAG



### 3.1.2 Operační systém

Jelikož je zařízení Raspberry Pi, jak bylo popsáno v minulé kapitole, standardní počítač, je nutné zvolit vhodný operační systém (OS). V době zpracování projektu byl pro architekturu ARM, respektive pro Raspberry Pi, dostupný OS „Raspbian“ nebo „ArchlinuxArm“. V průběhu psaní této práce přibyl k těmto dvěma OS od veze 17 i OS „Fedora“.

#### Raspbian

Raspbian je odvozenina systému GNU Linux Debian. Jedná se tedy o velice stabilní systém s podporou široké technické veřejnosti. Raspbian je, na rozdíl od ArchlinuxArm, sestavován pouze pro Raspberry Pi, což zvyšuje stabilitu systému. Systém je tedy vhodný pro nasazení do dlouhodobého provozu.

#### ArchlinuxArm

ArchlinuxArm je dostupný pro platformy ARMv5, ARMv6 a ARMv7. Jedná se o rychle se rozvíjející systém s podporou nejnovějších balíčků a aktualizací. Zmíněná vlastnost jej posouvá vývojově dopředu oproti systému Debian, ale na druhou stranu způsobuje lehkou nestabilitu.

Tato práce využívá program SNMP, přičemž v OS ArchlinuxArm je tento program obsažen v balíčku „net-snmp“. Při pokusu spustit snmpd (SNMP – daemon) dojde k ukončení programu s chybovým hlášením **výpis: 3.1**. Je patrné, že snmpd vyžaduje přístup k pci sběrnici, která na desce Raspberry Pi není.

**Výpis: 3.1:** Chybové hlášení snmpd

```
sudo journalctl -n
<1690 () 17:56
-- Logs begin at Thu, 1970-01-01 01:00:02 BST, end at Thu, 1970-01-01 01:00:16
BST. --
Dec 09 17:56:47 raspberry_lukas sudo[297]: pam_unix(sudo:session): session opened
for user root by lukas(uid=0)
Dec 09 17:56:47 raspberry_lukas systemd[1]: Starting Simple Network Management
Protocol (SNMP) Daemon...
Dec 09 17:56:47 raspberry_lukas snmpd[300]: pcilib: Cannot open /proc/bus/pci
Dec 09 17:56:47 raspberry_lukas snmpd[300]: pcilib: Cannot find any working access
method.
Dec 09 17:56:47 raspberry_lukas systemd[1]: snmpd.service: control process exited,
code=exited status=1
Dec 09 17:56:47 raspberry_lukas systemd[1]: Failed to start Simple Network
Management Protocol (SNMP) Daemon.
Dec 09 17:56:47 raspberry_lukas systemd[1]: Unit snmpd.service entered failed state
Dec 09 17:56:47 raspberry_lukas sudo[297]: pam_unix(sudo:session): session closed
for user root
Dec 09 17:56:59 raspberry_lukas sudo[304]: lukas : TTY=pts/0 ; PWD=/home/lukas ;
USER=root ; COMMAND=/usr/bin/journalctl -n
```

```
Dec 09 17:57:00 raspberry_lukas sudo[304]: pam_unix(sudo:session): session opened
for user root by lukas(uid=0)
```

Tuto systémovou chybu se nepodařilo opravit ani vyjmutím podpory „pci“ při kompilaci SNMP. Některé internetové zdroje uvádí tuto možnost jako správnou, ale i v případě úspěšné rekompilace je nasazení takto upraveného balíčku (do zařízení dohledu a správy síťových prvků) neakceptovatelnou variantou.

Pro dokončení tohoto projektu je snmpd nezbytné. Proto nebude zařízení provozováno s distribucí ArchlinuxArm, ale s distribucí Raspbian.

## Fedora 17

Fedora17 [14] byla uvolněna až v rámci práce na tomto projektu a nebyla tudíž při výběru operačního systému testována.

### 3.1.3 Ovládání GPIO

Raspberry Pi v kombinaci s GNU Linux umožňuje přistupovat k GPIO několika způsoby a pomocí několika programovacích a skriptovacích jazyků. Jedná se především o Python, bash, javu a C. Velice efektivní je přístup k GPIO přímo z příkazové řádky. V systémové cestě `/sys/class/gpio` je namapována celá periferie GPIO. Drobnou nevýhodou je možnost přistupovat do `/sys/class/gpio` pouze pod uživatelem root. Zápis se provádí pomocí příkazu „echo“. Logickou hodnotu „H“ lze na výstup zapsat pomocí posloupnosti příkazů, výpis: 3.2.

**Výpis: 3.2:** Nastavení logické hodnoty „H“ na pinu 7

```
cd /sys/class/gpio          #přepnutí do hlavní sloky gpio
sudo echo "7" >> export      #aktivace pinu GPIO7
cd gpio7                    #přepnutí do hlavní složky pinu 7
sudo echo "out" >> direction #zapsání směru komunikace pro pin 7
sudo echo "1" >> value        #aktivace logické úrovně H na pinu 7
cat value                   #vyčtení hodnoty pinu 7
```

Obdobným způsobem fungují ekvivalentní hodnoty „in“, „0“ a také zápis do souboru „unexport“.

## Podpora GPIO

Projekt WiringPi [18] zavádí do systému podporu GPIO tak, aby bylo možné přistupovat k GPIO přímo bez použití uživatele root. Projekt WiringPi zavádí do systému knihovny pro programování v jazyce C.

Ve výpis: 3.3 je uveden postup stažení, sestavení a instalace projektu WiringPi [18]. Pro správnou funkci je nutné nainstalovat podporu pro program git. Úspěšná instalace je zakončena výstupem výpis: 3.4.

### Výpis: 3.3: Instalace podpory GPIO WiringPi

```
git clone git://git.drogon.net/wiringPi      #stažení poslední verze z~git archivu
cd wiringPi                                  #přepnutí do složky wiringPi
git pull origin                               #ověření aktuálnosti archívu
./build                                       #sestavení balíčku
```

### Výpis: 3.4: Konec výstupu instalátoru WiringPi

```
Examples
wiringPi Examples
There are now too many examples to compile them all in a sensible time,
and you probably don't want to compile or run them all anyway, so they
have been separated out.
To compile an individual example, just type
    make exampleName
Where exampleName is one of:
    test1 test2 speed lcd wfi piface gertboard nes delayTest softPwm
All Done.
```

Z výstupu instalace výpis: 3.4 je patrné, že součástí projektu je i několik vzorových příkladů pro programování v jazyce C.

## Použití WiringPi

Možnosti použití projektu WiringPi, respektive příkazu „gpio“, jsou vyobrazeny ve výpis: 3.5. Ve výpis: 3.2 bylo uvedeno nastavení pinu 7 do úrovně „H“, v odstavci výpis: 3.6 je uveden ekvivalentní zápis pomocí příkazu „gpio“.

### Výpis: 3.5: Nápověda k příkazu gpio

```
Usage: gpio -v
gpio -h
gpio [-g] <read/write/pwm/mode> ...
gpio [-p] <read/write/mode> ...
gpio readall
gpio unexportall/exports ...
gpio export/edge/unexport ...
gpio drive <group> <value>
gpio pwm-bal/pwm-ms
gpio pwmr <range>
gpio pwmc <divider>
gpio load spi/i2c
gpio gbr <channel>
gpio gbw <channel> <value>
```

### Výpis: 3.6: Nastavení logické hodnoty „H“ na pinu 7

```
gpio export 7 out
gpio -g write 7 1
gpio -g read 7
```

## 3.2 Měřicí modul

Ke zmíněnému mikropočítači Raspberry Pi bylo v rámci tohoto projektu nutné navrhnout a vyrobit rozšiřující desku umožňující měřit spotřebu elektrické energie. V rámci vývoje byly navrženy dvě verze rozšiřujícího modulu.

### 3.2.1 Měřicí modul – první verze

První verze byla připojena k Raspberry Pi pomocí dvouřadého 26 pinového rozhraní obsahujícího jak GPIO, tak napájení 5 V a 3,3 V. V této verzi rozšiřujícího modulu byly obsaženy všechny prvky potřebné k funkci na jedné DPS. Jednalo se zejména o tyto prvky: měřicí obvod ADE7763, optické oddělení (pouze SPI sběrnice), DC | DC měnič a bistabilní relé.

U této vývojové verze se projevíly následující nedostatky:

1. Chybějící optické oddělení pinů ovládajících relé.
2. Chybějící ochrana pinů Raspberry Pi.
3. Chybějící měřicí body.
4. Chybějící signalizační LED diody.
5. Prosakování 230 V na piny Raspberry Pi.
6. Nepoužitelné rozměry desky.

### 3.2.2 Obvod ADE7763

Pro měření spotřeby elektrické energie a převod hodnot do digitální podoby lze využít jednoúčelového obvodu. Z tabulky 1.1, respektive z nabídky firmy ANALOG DEVICES, byl pro realizaci tohoto projektu vybrán obvod ADE7763 [2]. Při výběru byl brán ohled především na typ komunikačního rozhraní a typ pouzdra. Obvod disponuje komunikační sběrnicí SPI a pulzním výstupem, jak je vidět v tabulce 1.1. Pro účely tohoto projektu bude použita komunikace SPI. Rozhraní obvodu ADE7763 [2] je připojeno přes optické oddělení (ADuM1401) na desku Raspberry Pi, konkrétně na piny 19, 21, 23 a 26. Na těchto pinech je hardwarová podpora SPI (Raspberry Pi model B verze 1). Celkové zapojení obvodu ADE7763 bylo provedeno podle doporučení výrobce a dostupné dokumentace [2]. Zapojení obvodu ADE7763 je zobrazeno v příloze, obr. D.3.

### 3.2.3 Optické oddělení

Měřicí obvod je galvanicky spojen přímo se síťovým napětím 230 V. Takovéto napětí není možné přímo přivést na piny Raspberry Pi. Pomocí optického oddělovače ADuM1401 [2] je zamezeno proniknutí nebezpečného napětí do obvodů Raspberry Pi. Zapojení obvodu ADuM1401 bylo provedeno podle doporučení výrobce [2] a je vyobrazeno na schématu obr. D.2.

### 3.2.4 DC | DC měnič

Část zařízení za optickým oddělením a samotný optický oddělovač potřebuje samostatné napájení, aby nemohlo dojít ke vniknutí napětí 230 V na desku Raspberry Pi. Bylo ověřeno, že lze provádět napájení přímo z 230 V bez použití transformátoru, pomocí svitkového kondenzátoru, stabilizátoru a diod, obr. D.1 [5]. Zatížitelnost „transformerless“ zdroje je velice malá a nesplňuje výkonové nároky optického oddělovače. Zavádět externí napájení by bylo značně neefektivní, proto bylo napájení vyřešeno pomocí DC | DC měniče. Jedná se o napěťový měnič 5 V na 5 V. Díky tomuto měniči je možné celý obvod napájet přímo z desky Raspberry Pi. Zároveň je zajištěno bezpečné oddělení síťového napětí 230 V. Zapojení DC | DC měniče je v příloze, obr. D.2.

### 3.2.5 Bistabilní relé

Zařízení je navrženo tak, aby bylo schopno odstavit měřený okruh síťového napětí. K odstavení může dojít automaticky při přetížení měřeného okruhu, nebo vzdáleně uživatelem. Rozpojení obvodu je prováděno bistabilním relé, které je připojeno pomocnými PNP tranzistory na piny Raspberry Pi GPIO 23 a GPIO 24. Bistabilní relé bylo zvoleno z důvodů udržení nízké provozní spotřeby Raspberry Pi. Zapojení bistabilního relé je v příloze, obr. D.2

### 3.2.6 Měřicí modul – druhá verze

Druhá verze měřicího modulu odstraňuje chyby zmíněné u první verze. Hlavní důraz byl kladen na ochranu Raspberry Pi. Tranzistory ovládající bistabilní relé byly nahrazeny optickým oddělením, obr. G.2, a každý pin Raspberry Pi byl doplněn o přepěťovou ochranu vyřešenou pomocí transilů, obr. G.3. (V rámci práce na projektu byla tato přepěťová ochrana nechtěně otestována, přičemž se projevila jako funkční a namísto. Modul Raspberry Pi nebyl poškozen, kdežto měřicí modul shořel kompletně.)

Druhá verze měřicího modulu zavádí změnu také v koncepci DPS. Návrh DPS, obr. H.1 a obr. H.2, byl rozdělen do několika částí – „pater“. První patro obsahuje prvky společné pro následující části. Jedná se o přepětovou ochranu (transily) a DC | DC měnič. Tato DPS je k Raspberry Pi připojena stejným způsobem jako první verze měřicího modulu, a to dvouřadým 26 pinovým dutinkovým konektorem. Ostatní moduly jsou připojeny k této části vždy dvěma jednořadými dutinkovými konektory umístěnými naproti sobě, po stranách DPS. Tato „sendvičová“ koncepce zajišťuje dobrou mechanickou stabilitu a umožňuje libovolně připojovat a odpojovat jeden nebo více rozšiřujících modulů.

Mechanické pevnosti lze dosáhnout využitím montážních otvorů, kompatibilních s Raspberry Pi (Model B veze 2) a dvou montážních otvorů určených k upevnění do případné krabice, nebo na panel. Náhled na DPS obr. H.4.

Náhled obou verzí je v příloze, obr. F.1 a I.1.

### 3.2.7 Náměty pro následující revizi

- **Napájení**

Celé zařízení je napájeno z micro USB konektoru umístěného na Raspberry Pi. Toto řešení je pro aplikaci v „terénu“ nevhodné. Zařízení bude použito v prostředí síťových prvků. Bylo by vhodné přizpůsobit napájení standardu POE.

- **CS**

Raspberry Pi – jeho HW SPI disponuje pouze dvěma CS. Tato vlastnost omezuje zařízení pouze na dva rozšiřující výkonové moduly. Rozšířit moduly o CS expander.

- **Svorkovnice 230V**

Vyměnit za výkonový konektor odpojitelný bez nástrojů.

- **Relé – signalizace**

Využít druhou svorku relé pro signalizaci stavu. Jelikož se jedná o bistabilní relé, nelze bez zpětné vazby určit v jaké je poloze.

- **Relé – výkonové požadavky**

Změnit typ relé na proudově odolnější.

- **Šumové jevy na bočníku**

Zaměřit se na snížení šumu na bočniku. Využít kvalitnější rezistory. Využít diferenciálních vodičů pro připojení bočniku. Omezit délku cest na DPS vedoucích k bočniku.

- **„Výkonová“ LED dioda**

Přidat LED diodu na CF out ADE7763 (frekvenční modulace). Indikace podobná jako na domovních hodinách. Zjistit, zdali je to možné.

- **Popisky**

Doplnit desku o popisky konektorů. (Na obou vrstvách.)

- **Indikační LED dioda**

Přidat LED indikující napájení za DC | DC měničem. (Odpojitelná jumperem.)

- **Ochrana převodníků ADE7763**

Převodníky ADE7763 mají maximální vstupní napětí 0,5 V. Při měření může dojít vlivem připojení nebo odpojením zátěže k záskmitu většímu než 0,5 V, obr. 6.15. Návrh je potřeba doplnit o přepětovou ochranu.

- **Zrcadlový návrh**

V rámci návrhu DPS byl použit nový program. Deska plošných spojů je funkční, ale navržená zrcadlově – je nutné i součástky osadit zrcadlově. V následující verzi je nutné opravit návrh DPS.

## 4 IMPLEMENTACE PROTOKOLŮ

### 4.1 Implementace protokolu SNMP

Jak bylo výše zmíněno v kapitole věnované výběru operačního systému, kap. 3.1.2, je implementace protokolu SNMP možná pouze pro OS Raspbian. V OS ArchlinuxArm je balíček `net-snmp` dostupný také, ale daemon `snmpd` nelze spustit běžným způsobem. V operačním systému ArchlinuxArm je totiž ponechána podpora pci sběrnice, kterou Raspberry Pi nevlastní. Spuštění daemona, výpis: 4.1, skončí chybovým hlášením, výpis: 4.2. Příkazem `journalctl`, výpis: 4.3, lze zobrazit dostupná chybová hlášení, výpis: 4.4. V tomto výpisu se nalézají již zmíněné chybové hlášení ohledně nedostupné pci sběrnice. Tuto chybu se v rámci této práce nepodařilo ze systému odstranit.

#### Výpis: 4.1: Aktivace daemona snmpd

```
sudo systemctl start snmpd.service
```

#### Výpis: 4.2: Chybové hlášení systemctl

```
Job for snmpd.service failed. See 'systemctl status snmpd.service' and 'journalctl -n' for details.
```

#### Výpis: 4.3: Vypsání chybových hlášení pro systemctl

```
sudo journalctl -n
```

#### Výpis: 4.4: Chybové hlášení pro spuštění snmpd

```
-- Logs begin at Thu, 1970-01-01 01:00:02 BST, end at Thu, 1970-01-01 01:00:16 BST. --
Dec 01 11:05:00 raspberry_lukas sudo[320]: pam_unix(sudo:session): session opened for user root by lukas(uid=0)
Dec 01 11:05:00 raspberry_lukas systemd[1]: Starting Simple Network Management Protocol (SNMP) Daemon...
Dec 01 11:05:01 raspberry_lukas snmpd[323]: pcilib: Cannot open /proc/bus/pci
Dec 01 11:05:01 raspberry_lukas snmpd[323]: pcilib: Cannot find any working access method.
Dec 01 11:05:01 raspberry_lukas systemd[1]: snmpd.service: control process exited, code=exited
status=1
Dec 01 11:05:01 raspberry_lukas systemd[1]: Failed to start Simple Network Management Protocol (SNMP) Daemon.
Dec 01 11:05:01 raspberry_lukas systemd[1]: Unit snmpd.service entered failed state
Dec 01 11:05:01 raspberry_lukas sudo[320]: pam_unix(sudo:session): session closed for user root
Dec 01 11:05:10 raspberry_lukas sudo[325]: lukas : TTY=pts/0 ; PWD=/home/lukas ; USER=root ;
```



```
COMMAND=/usr/bin/journalctl -n
Dec 01 11:05:10 raspberry_lukas sudo[325]: pam_unix(sudo:session): session opened
      for user root by
lukas(uid=0)
```

### 4.1.1 Instalace SNMP

Instalace SNMP podle webové podpory OS Debian [22].

#### Instalace serveru snmpd

Nejdříve byla provedena instalace serveru snmpd, **výpis: 4.5**. Po instalaci je vhodné vyexportovat MIB, **výpis: 4.6**, a provést restart serveru, **výpis: 4.7**, který by měl proběhnout bez chybového hlášení.

##### Výpis: 4.5: Instalace snmpd

```
apt-get install snmpd #root
```

##### Výpis: 4.6: Export MIB

```
export MIBS=/usr/share/mibs
```

##### Výpis: 4.7: Restart snmpd

```
/etc/init.d/snmpd restart #root
```

#### Rozbor konfiguračního souboru snmpd.conf

Hlavní konfigurační soubor pro snmpd se nachází v `/etc/snmp/snmpd.conf`. Podle dokumentace OS Fedora [9] byl sestaven následující konfigurační soubor `snmpd.conf`. Celý konfigurační soubor se nachází v příloze, **výpis: L.1**.

### 4.1.2 Konfigurace SNMPv2c

#### Přístupová pravidla SNMP Version 2c Community

Verze protokolu 2 podporuje řízení přístupu pro různé uživatele. Zápis je prováděn do souboru `snmpd.conf` v pořadí:

##### directive community source OID

Po editaci konfiguračního souboru je nutné provést restart snmpd, **výpis: 4.7**.

- **directive** [rocommunity | rwcommunity]
  - Určuje, zda se jedná a komunitu ReadOnly nebo Read and Write přístup.

- **community** [name]
  - Jedná se o název community – uživatelské jméno.
- **source** [IP]
  - Omezení přístupu pro jeden jediný PC – jednu jedinou IP adresu.
- **OID**
  - Jedná se o zpřístupněné větve stromu MIB.

**Výpis: 4.8:** Příklad ReadOnly community na adrese 192.168.1.10, SNMPv2c

#directive	community	source	OID
rocommunity	read_only_user	192.168.1.10	.1.3.6

## Test přístupu pro Version 2c Community

Ze vzdáleného počítače je možné provést vyčtení systémových informací pomocí příkazu `snmpwalk`, výpis: 4.9.

**Výpis: 4.9:** Příklad načítání systémových dat z agenta na adrese ...15, SNMPv2c

```
snmpwalk -v2c -c read_only_user 192.168.1.15 system
```

Parametry příkazu `snmpwalk`:

- **-v2c**
  - Verze SNMP protokolu.
- **-c**
  - COMMUNITY (uživatelské jméno).
- **192.168.1.10**
  - Adresa agenta.
- **system**
  - Část MIB záznamu.

**Výpis: 4.10:** Příklad odpovědi agenta na dotaz výpis: 4.9, SNMPv2c

```
SNMPv2-MIB::sysDescr.0 = STRING: Linux raspberrypi 3.2.27+ #250 PREEMPT Thu Oct 18
19:03:02 BST 2012
armv6l
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (607686) 1:41:16.86
SNMPv2-MIB::sysContact.0 = STRING: UNIX Admin <admin@example.com>
SNMPv2-MIB::sysName.0 = STRING: raspberrypi
```

```

SNMPv2-MIB::sysLocation.0 = STRING: Datacenter, Row 3, Rack 2
SNMPv2-MIB::sysServices.0 = INTEGER: 76
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (4) 0:00:00.04
SNMPv2-MIB::sysORID.1 = OID: SNMP-FRAMEWORK-MIB::snmpFrameworkMIBCompliance
SNMPv2-MIB::sysORID.2 = OID: SNMP-MPD-MIB::snmpMPDCompliance
SNMPv2-MIB::sysORID.3 = OID: SNMP-USER-BASED-SM-MIB::usmMIBCompliance
SNMPv2-MIB::sysORID.4 = OID: SNMPv2-MIB::snmpMIB
SNMPv2-MIB::sysORID.5 = OID: TCP-MIB::tcpMIB
SNMPv2-MIB::sysORID.6 = OID: IP-MIB::ip
SNMPv2-MIB::sysORID.7 = OID: UDP-MIB::udpMIB
SNMPv2-MIB::sysORID.8 = OID: SNMP-VIEW-BASED-ACM-MIB::vacmBasicGroup
SNMPv2-MIB::sysORDescr.1 = STRING: The SNMP Management Architecture MIB.
SNMPv2-MIB::sysORDescr.2 = STRING: The MIB for Message Processing and Dispatching.
SNMPv2-MIB::sysORDescr.3 = STRING: The management information definitions for the
    SNMP User-based
    Security Model.
SNMPv2-MIB::sysORDescr.4 = STRING: The MIB module for SNMPv2 entities
SNMPv2-MIB::sysORDescr.5 = STRING: The MIB module for managing TCP implementations
SNMPv2-MIB::sysORDescr.6 = STRING: The MIB module for managing IP and ICMP
    implementations
SNMPv2-MIB::sysORDescr.7 = STRING: The MIB module for managing UDP implementations
SNMPv2-MIB::sysORDescr.8 = STRING: View-based Access Control Model for SNMP.
SNMPv2-MIB::sysORUpTime.1 = Timeticks: (3) 0:00:00.03
SNMPv2-MIB::sysORUpTime.2 = Timeticks: (4) 0:00:00.04
SNMPv2-MIB::sysORUpTime.3 = Timeticks: (4) 0:00:00.04
SNMPv2-MIB::sysORUpTime.4 = Timeticks: (4) 0:00:00.04
SNMPv2-MIB::sysORUpTime.5 = Timeticks: (4) 0:00:00.04
SNMPv2-MIB::sysORUpTime.6 = Timeticks: (4) 0:00:00.04
SNMPv2-MIB::sysORUpTime.7 = Timeticks: (4) 0:00:00.04
SNMPv2-MIB::sysORUpTime.8 = Timeticks: (4) 0:00:00.04

```

Pro zkrácení výpisu je možné zažádat agenta pouze o konkrétní řádek, jak je tomu ve výpis: 4.11.

**Výpis: 4.11:** Příklad žádosti agenta o jeden řádek záznamu, SNMPv2c

```

snmpwalk -v2c -c read_only_user 192.168.1.15 sysUpTime
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (757791) 2:06:17.91

```

### 4.1.3 Konfigurace SNMPv3

#### Přístupová pravidla SNMP Version 3

SNMP ve verzi 3 vyžaduje konfiguraci jednotlivých uživatelů. Provádí se při zastaveném snmpd. OS Archlinux má pro přidání uživatelů příkaz `net-snmp-create-v3-user`. Ten však není v distribuci Raspbian k dispozici a konfigurace se provádí ručně. V souboru `/etc/snmp/snmpd.conf` je potřeba vytvořit uživatele a jeho práva, výpis: 4.13. Dále je potřeba v souboru `/var/lib/snmp/snmpd.conf` přidat řádek, který nového uživatele při spuštění snmpd vytvoří, výpis: 4.14. Pokud je soubor `/var/lib/snmp/snmpd.conf` editován při běžícím snmpd, jsou veškeré změny po zastavení snmpd, výpis: 4.12, odstraněny. Příkaz pro vytvoření nového uživatele se přidává na

konec souboru `/var/lib/snmp/snmpd.conf`. Po obnovení běhu `snmpd`, výpis: 4.15, je tento poslední řádek odstraněn a přibližně uprostřed souboru je vygenerován otisk uživatele.

**Výpis: 4.12:** Zastavení běhu `snmpd`.

```
sudo /etc/init.d/snmpd stop
```

**Výpis: 4.13:** Příklad přidělení práv uživateli `snmpd` `/etc/snmp/snmpd.conf`, SNMPv3.

```
###Configuring SNMP Version 3
#directive user [rwuser|rouser] user name Security Levels in SNMPv3 [OID]
rwuser userV3RO authPriv .1.3.6
```

Popis konfigurační struktury:

- **directive user** `[rwuser | rouser]`
  - Nastavení práv uživatele na Read/Write a na Rad only.
- **user name**
  - Uživatelské jméno.
- **Security Levels in SNMPv3** `[noAuthnoPriv | authNoPriv | authPriv]`
  - Způsob ověření uživatele.
- **OID**
  - Zpřístupněná úroveň MIB stromu.

**Výpis: 4.14:** Příklad posledních tří řádků souboru `/var/lib/snmp/snmpd.conf`. Poslední řádek přidá nového uživatele.

```
engineBoots 10
oldEngineID 0x80001f8880ffa37908bcb2b850
createUser userV3RO MD5 "12345678" DES 12345678
```

Popis konfigurační struktury:

- **createUser**
  - Příkaz pro vytvoření uživatele.
- **userV3RO**
  - Uživatelské jméno.
- **MD5 "12345678"**
  - Kontrolní součet pro heslo 12345678 – fingerprint.

- **DES 12345678**

- Šifrovací algoritmus a heslo 12345678 – symetrický šifrovací algoritmus.

- Heslo je záměrně uvedeno "12345678". Pro správnou funkci musí být heslo minimálně **osmi** místné.

**Výpis: 4.15:** Obnovení běhu snmpd.

```
sudo /etc/init.d/snmpd start
```

## Konfigurace klienta

Pro připojení klienta pomocí protokolu SNMP, **výpis: 4.17**, je vhodné vytvořit konfigurační soubor `~/snmp/snmp.conf` s obsahem odpovídajícím předchozímu nastavení, **výpis: 4.16**.

**Výpis: 4.16:** Příklad konfiguračního souboru `~/snmp/snmp.conf`.

```
defVersion 3
defSecurityLevel authPriv
defSecurityName userV3R0
defPassphrase 12345678
```

**Výpis: 4.17:** Příklad žádosti agenta o systémové informace, SNMPv3

```
snmpwalk -v3 192.168.1.15 system
```

## 4.1.4 Další možnosti nastavení SNMP

### Příkazy pro shell pomocí extended

Konfigurační soubor `/etc/snmp/snmpd.conf` umožňuje pomocí příkazu `extended` spustit jakýkoliv vnitřní příkaz systému. Sem se řadí například spustitelný skript, binární soubor nebo alias. Zápis se provádí ve tvaru „`extend name path_to_file parameters`“. Na místě spustitelného souboru musí být absolutní cesta, jak je uvedeno v příkladu, **výpis: 4.18**. V případě záznamu `extend`, **výpis: 4.18**, v souboru `/etc/snmp/snmpd.conf` je odpověď na žádost `snmpwalk`, **výpis: 4.19**, zobrazena ve **výpis: 4.20**.

**Výpis: 4.18:** Příklad zápisu `extend` pro příkaz `date`

```
# extend    name          path_to_executable_file    parameters
extend     datum         /usr/bin/date
```

**Výpis: 4.19:** Příklad žádosti agenta o vypsání `extend` položek

```
snmpwalk -v3 192.168.1.15 NET-SNMP-EXTEND-MIB::nsExtendOutLine
```

#### Výpis: 4.20: Příklad odpovědi agenta na dotaz výpis: 4.19

```
NET-SNMP-EXTEND-MIB::nsExtendOutLine."datum".1 = STRING: P.. pro 7 12:41:57 CET
201
```

Pokud je spouštěný příkaz skriptem, je nutné zápis upravit v `/etc/snmp/snmpd.conf` do podoby, výpis: 4.21, kde spustitelný soubor je `/bin/sh`, `/bin/bash` nebo jiný příkazový interpret a spouštěný skript je jeho parametrem. Parametry se dají řetězit, takže za cestou ke skriptu může následovat vstupní parametr pro skript.

První vstupní proměnná skriptu se načítá automaticky do `$1` a analogicky ostatní proměnné do `$x`. Spouštěný skript musí obsahovat výstupní data, jak je uvedeno ve výpis: 4.21 pomocí `echo` nebo návratové hodnoty `exit $out`. Pokud je výstupní hodnota číslo, nikoliv řetězec, je vypisována jako integer. Je nutné brát ohled na rozsah integeru **0 – 255**.

#### Výpis: 4.21: Příklad zápisu extend pro spuštění skriptu

```
# extend name path_to_executable_file parameter 1 parameter 2
extend cislo /bin/sh /home/lukas/skripts/test_snmp.sh 5
```

#### Výpis: 4.22: Příklad skriptu pro výpis: 4.21

```
#!/bin/sh

case $1 in
1) out=11111111;;
2) out=123;;
3) out=255;;
4) out=256;;
5) out=257;;
6) out=66666666;;
7) out=77777777;;
8) out=88888888;;
esac

echo "Prvni_radek_vystupu_cislo_je_$out."

exit $out
```

### Informace o zařízení

Pro soubor `/etc/snmp/snmpd.conf` jsou definovány tyto záznamy:

- **syslocation** Datacenter, Row 3, Rack 2

Umístění zařízení – textový řetězec.

- **syscontact** UNIX Admin admin@example.com

Kontakt na administrátora – textový řetězec.

- **syssservices** 76

Servisní číslo – numerické hodnota.

## Informace o discích a procesech

Do souboru `/etc/snmp/snmpd.conf` lze zapsat záznamy s klíčovými slovy **proc** a **disk**, výpis: 4.23, pro zjištění stavu monitorovaných procesů a informace o discích. Záznamy **proc** a **disk** mají kontrolní parametry. Při překročení těchto referenčních hodnot může SNMP vyvolat chybový stav a informovat správce pomocí trap-ů.

**Výpis: 4.23:** Příklad zápisu **proc** a **disk** v `/etc/snmp/snmpd.conf`

```
proc    sshd 1 1
disk    /tmp 40000
```

- **proc** proces MAX MIN

Kontrola běžících procesů.

Konkrétní proces (sshd).

MAX maximální povolený počet běžících procesů – výchozí hodnota 0.

MIN minimální počet běžících procesů – výchozí hodnota 0.

- **disk** přípojný\_bod volné\_místo

Monitorování připojených disků.

Přípojný bod `/tmp`, výpis: 4.23.

Velikost volného místa – výchozí hodnota 100000 – zadává se v [Mib's]

## 4.2 Implementace ostatních protokolů

### 4.2.1 Protokol SSH

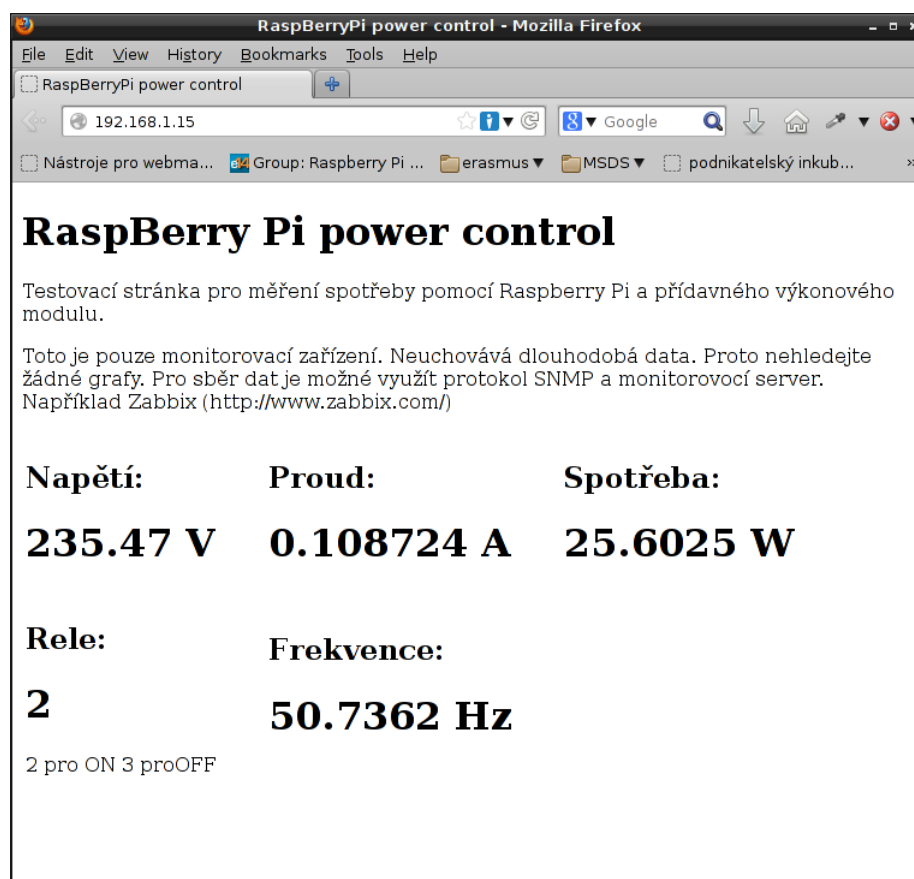
Dalším aplikovaným protokolem pro řízení, správu a dohled je protokol SSH. SSH je standardní výbavou většiny serverů. Základním rozhraním pro servery typu UNIX a GNU Linux je příkazový řádek. SSH poskytuje zabezpečený vzdálený přístup do příkazového řádku, a tím i k celému systému. Dává uživateli velice komplexní přístup k zařízení. Umožňuje vzdáleně přistoupit na Raspberry Pi a převzít tak kompletní správu zařízení. Z tohoto důvodu implementace SSH protokolu nesmí chybět. SSH je standardním balíčkem všech GNU Linux OS. Instalace a nastavení je rutinní operací a nebude zde komentována.

## 4.2.2 Protokol HTTP

Jak bylo již zmíněno, Raspberry Pi lze řadit mezi plnohodnotné počítače. Svým výkonem sice nemůže konkurovat stolním nebo serverovým počítačům, ale s nainstalovaným OS GNU Linux lze i na takto malém zařízení bez větších problémů zprovoznit webový server. Webový server sice není hlavním zájmem této práce, ale pro kompletní využití možností výsledného zařízení byla vytvořena jednoduchá webová stránka s nejdůležitějšími informacemi o měření, náhled stránky obr. 4.1.

Instalace webového serveru Apache na Raspberry Pi byla provedena podle návodu [25] na stránkách Ubuntu – jedná se rovněž o klon systému Debian tak, jako Raspbian provozovaný na Raspberry Pi. Instalace mysql, jak je uvedena v návodě, je pro funkci zařízení zbytečná a samotný běh mysql serveru pohltí velké množství systémových prostředků. Proto instalace mysql na Raspberry Pi není doporučena, pokud to není nezbytně nutné.

Zdrojový kód stránky je zobrazen v příloze výpis: L.2.



**Obr. 4.1:** Náhled na webovou stránku umístěnou na webovém serveru Apache spuštěném přímo na Raspberry Pi



## 5 PŘENÁŠENÁ DATA, PERIODA SBĚRU DAT A LOKÁLNÍ ZÁLOHOVÁNÍ

### 5.1 Velikost přenášených dat

Hlavní funkcí konstruovaného zařízení je poskytovat neustálý přehled o stavu měřeného zařízení. Nejdůležitější měřenou veličinou je elektrický výkon, respektive spotřeba elektrické energie. Další možnou poskytovanou veličinou je průměrná hodnota spotřeby elektrické energie. Měřený okruh je osazen spínacím relé, pomocí kterého je možné odstavit celý měřený okruh od síťového napětí. Je tedy nutné přenášet informaci i o stavu tohoto relé.

Pomocí protokolu SNMP zmíněného v kapitole 4.1 lze z agenta vyčíst potřebné informace o stavu zařízení **výpis: 5.1**, konkrétně stav relé „rele“, aktuální spotřebu elektrické energie „power“ a průměrnou hodnotu spotřeby elektrické energie „avg-power“ jak je uvedeno ve **výpis: 5.2**.

**Výpis: 5.1:** Žádost SNMP agenta o zaslání informací o spotřebě.

```
snmpwalk -v2c -c read_only_user 192.168.1.10 NET-SNMP-EXTEND-MIB::nsExtendOutLine
```

**Výpis: 5.2:** Odpověď agenta SNMP na žádost o zaslání informací o spotřebě.

```
NET-SNMP-EXTEND-MIB::nsExtendOutLine."rele".1 = STRING: 1
NET-SNMP-EXTEND-MIB::nsExtendOutLine."power".1 = STRING: 40.83
NET-SNMP-EXTEND-MIB::nsExtendOutLine."avg_power".1 = STRING: 36.59
```

**Výpis: 5.3:** Výstup monitorovacího programu Wireshark.

No.	Time	Source	Destination	Protocol	Length
490	2.867081000	192.168.1.10	192.168.1.15	SNMP	98
491	2.888492000	192.168.1.15	192.168.1.10	SNMP	105
496	2.891923000	192.168.1.10	192.168.1.15	SNMP	104
497	2.912833000	192.168.1.15	192.168.1.10	SNMP	110
502	2.915423000	192.168.1.10	192.168.1.15	SNMP	105
503	2.937428000	192.168.1.15	192.168.1.10	SNMP	114
508	2.940043000	192.168.1.10	192.168.1.15	SNMP	109
509	2.940481000	192.168.1.15	192.168.1.10	SNMP	97

V části **výpis: 5.3** je zobrazen odfiltrovaný výstup síťového monitorovacího programu Wireshark. Zachycené rámce odpovídají komunikaci „klient – agent“ pro zmíněné příkazy **výpis: 5.1** a **výpis: 5.2**. Celá relace, jak je vidět z **výpis: 5.3**, zabrala 842 byte. Při využití protokolu SNMPv3 celá komunikace zabere dva rámce o celkové délce 256 byte.

## 5.2 Zatížení klienta a agenta při čtení dat

Pro kompletní představu o vytížení obou systémů (monitorovaný a monitorující) je na následujících obrázcích zachycen stav systému při monitorování.

Pomocí SNMP jsou dostupné informace o napětí, proudu, frekvenci a výkonu. Pro kompletní měření musí být na straně monitorovacího serveru spuštěny čtyři dotazy `snmpwalk`. Tato činnost byla při měření řešena nekonečným cyklem napsaným ve skriptu `read_snmp_data`. Zatížení systému tímto `bash` skriptem, obr. 5.2, se pohybovalo kolem 0,3 % CPU.

Samotné čtení dat příkazem `snmpwalk`, obr. 5.1, se pohybovalo do 6,5 % CPU (po velice krátkou dobu).

Na druhé straně, vytížení Raspberry Pi měřicím programem, obr. 5.3, se pohybovalo stabilně kolem 3 % CPU. Aktivita SNMP agenta nebyla pro jeho nenáročnost na systémové prostředky zaznamenána. Jeho náročnost se projevuje pouze na využití RAM paměti, na obr. 5.3 je vidět obsazení 2,1 % paměti daemonem `snmpd`.

```
top - 23:05:56 up 23:13, 2 users, load average: 0.04, 0.03, 0.00
Tasks: 88 total, 2 running, 86 sleeping, 0 stopped, 0 zombie
Cpu(s): 40.0%us, 3.3%sy, 0.0%ni, 56.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1034464k total, 464420k used, 570044k free, 120648k buffers
Swap: 688120k total, 0k used, 688120k free, 275072k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
29055	lukas	20	0	4588	2068	1372	R	6.5	0.2	0:00.02	snmpwalk
1	root	20	0	2028	700	608	S	0.0	0.1	0:01.64	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.32	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0

Obr. 5.1: Zatížení monitorovacího serveru při čtení SNMP dat

```
top - 23:09:26 up 23:17, 2 users, load average: 0.00, 0.01, 0.00
Tasks: 88 total, 1 running, 87 sleeping, 0 stopped, 0 zombie
Cpu(s): 4.0%us, 0.3%sy, 0.0%ni, 95.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1034464k total, 464392k used, 570072k free, 120720k buffers
Swap: 688120k total, 0k used, 688120k free, 275080k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2670	lukas	20	0	3132	1176	1016	S	0.3	0.1	0:21.22	read_snmp_data.
1	root	20	0	2028	700	608	S	0.0	0.1	0:01.64	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.32	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0

Obr. 5.2: Zatížení monitorovacího serveru při cyklickém spouštění příkazu `snmpwalk`

CPU[     ] 11.9%										Tasks: 39, 72 thr;	
Mem[     ] 49/184MB										Load average: 2.14	
Swp[ ] 0/99MB										Uptime: 23:23:53	
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1785	lukas	20	0	5616	2076	1244	R	6.0	1.1	0:03.14	htop
1822	root	20	0	3156	1140	976	D	3.0	0.6	44:48.01	PowerControll
2077	snmp	20	0	9372	4024	2608	S	0.0	2.1	7:35.93	/usr/sbin/snmpd -Lsd -Lf /dev/null
1845	root	20	0	28988	1572	1156	S	0.0	0.8	2:11.93	/usr/sbin/rsyslogd -c5
1844	root	20	0	28988	1572	1156	S	0.0	0.8	2:43.15	/usr/sbin/rsyslogd -c5
1	root	20	0	2140	724	616	S	0.0	0.4	0:04.45	init [2]
154	root	20	0	2876	1276	740	S	0.0	0.7	0:00.45	udev --daemon
274	root	20	0	2872	996	456	S	0.0	0.5	0:00.01	udev --daemon

Obr. 5.3: Zatížení Raspberry Pi měřícím programem

## 5.3 Perioda sběru dat

Perioda odečítání dat závisí především na požadované přesnosti měření, tedy na monitorovacím serveru a jeho požadavcích. Jak bylo popsáno v kapitole 5.1, je velikost přenášených dat velice malá. Pro kapacitu dnešních sítí téměř zanedbatelná. Navrhované zařízení tedy není omezeno kapacitou sítě a sběr dat může probíhat libovolně často.

Tato práce se zabývá měřením vzdálených, především síťových, prvků. Jedná se o malé odběry s téměř konstantním časovým průběhem, což umožňuje značně prodloužit periodu kontroly stavu odběru elektrické energie.

V rámci testovacího měření bylo použito intervalu 5 s. Tento časový úsek se jeví jako dostatečně krátký na zachycení podstatných informací a nedochází ani k zahlcení Raspberry Pi příliš častými dotazy.

## 5.4 Lokální zálohování

Naměřená data spotřeby elektrické energie je potřeba ukládat a zpracovávat. Hlavní podíl na zálohování dat a jejich zpracování má podle koncepčního řešení dohledový server, který má za úkol monitorovat více zařízení pomocí protokolu SNMP a shromažďovat informace o stavu sítě, popřípadě tyto informace a data vyhodnocovat. Není tedy nutné řešit dlouhodobou zálohu naměřených dat. Jistá záloha dat však probíhat musí.

Raspberry Pi má standardně jako jediné úložiště SD kartu, která podléhá omezené životnosti co se týče počtu cyklů zápisů. Podle dokumentace [20] společnosti Sandisk, která jako jedna z mála společností poskytuje manuálové stránky k paměťovým SD kartám, je garantovaná životnost 10000 zapisovacích cyklů. Je tedy nutné omezit přehnané zapisování na SD kartu.

### 5.4.1 RAM disk

Neustálé zapisování dat na SD kartu lze omezit vytvořením virtuálního paměťového prostoru v RAM paměti Raspberry Pi. Jedná se o alokaci RAM paměti do filesystému. Do takto vytvořeného úseku paměti lze zapisovat bez jakéhokoli omezení. Důležité je pouze data z RAM disku před vypnutím systému přesunout na paměťovou kartu. RAM paměť je „volatile“ a při odpojení od napájení ztrácí data. Jednou z možností jak připojit RAM disk je využití `/dev/ramX`. Nejdříve je potřeba příslušnou část RAM paměti naformátovat, **výpis: 5.4**, a následně připojit, **výpis: 5.5**.

**Výpis: 5.4:** Formátování části RAM paměti [17].

```
mkfs.ext3 /dev/ram0
```

**Výpis: 5.5:** Připojení části RAM paměti k filesystému [17].

```
mount /dev/ram0 /mnt/disk
```

Při využití RAM disku stačí data zálohovat na SD kartu pouze za účelem ochrany proti výpadku napájení a při vypnutí systému. Při maximálním počtu zápisů 10000 podle [20] a celoročním provozu se zápisem na SD kartu dvakrát denně, klesne životnost karty přibližně na 13 let provozu.

## 6 MĚŘENÍ

V následujících kapitolách bude popsán měřicí program, jeho automatické spouštění při startu systému a v závěru kapitoly budou prezentována data z celodenního testovacího měření.

### 6.1 Měřicí Software

Raspberry Pi se od klasického počítače liší v mnoha ohledech. Jak bylo již zmíněno v kapitole 3.1.3, Raspberry Pi umožňuje přímý přístup na periferie. Součástí této práce je program napsaný v programovacím jazyce C. Z programátorského hlediska se nejedná o nijak složitý kód, jeho struktura viz obr. 6.1.

Hlavní smyčka programu vykoná nejdříve inicializaci periférií, (binární soubor je nutné spouštět jako uživatel root). Následuje inicializace pracovního prostředí. Program kontroluje připojení RAM disku, viz kapitola 5.4.1. Při těchto operacích je využito funkce `system("systémová funkce")`. Jedná se konkrétně o aktivaci SPI periferie, naformátování RAM paměti a připojení RAM paměti na určené místo v systému. Využití tohoto zápisu omezuje přenositelnost výsledného kódu na jiné platformy, což ale není zásadním problémem, vezmeme-li v potaz, že momentálně existuje pro tuto aplikaci pouze jeden použitelný operační systém. Navržené měřicí moduly jsou také velice specifické a nepřenositelné.

Hlavní smyčka programu je zakončena nekonečným cyklem. V tomto cyklu je prováděno měření napětí, proudu, frekvence a výkonu<sup>1</sup>. Odečtená data jsou ukládána do textového souboru v RAM paměti.

Složitější než samotný kód jsou některé funkce, které program vykonává při samotném měření.

U obou veličin (napětí i proud) jsou data při rychlém odečítání značně nestálá. Tento problém byl ošetřen kruhovým bufferem, do kterého jsou ukládána nově získaná data a průměrována s několika staršími hodnotami dat. Délka kruhového bufferu byla nastavena na 100 prvků.

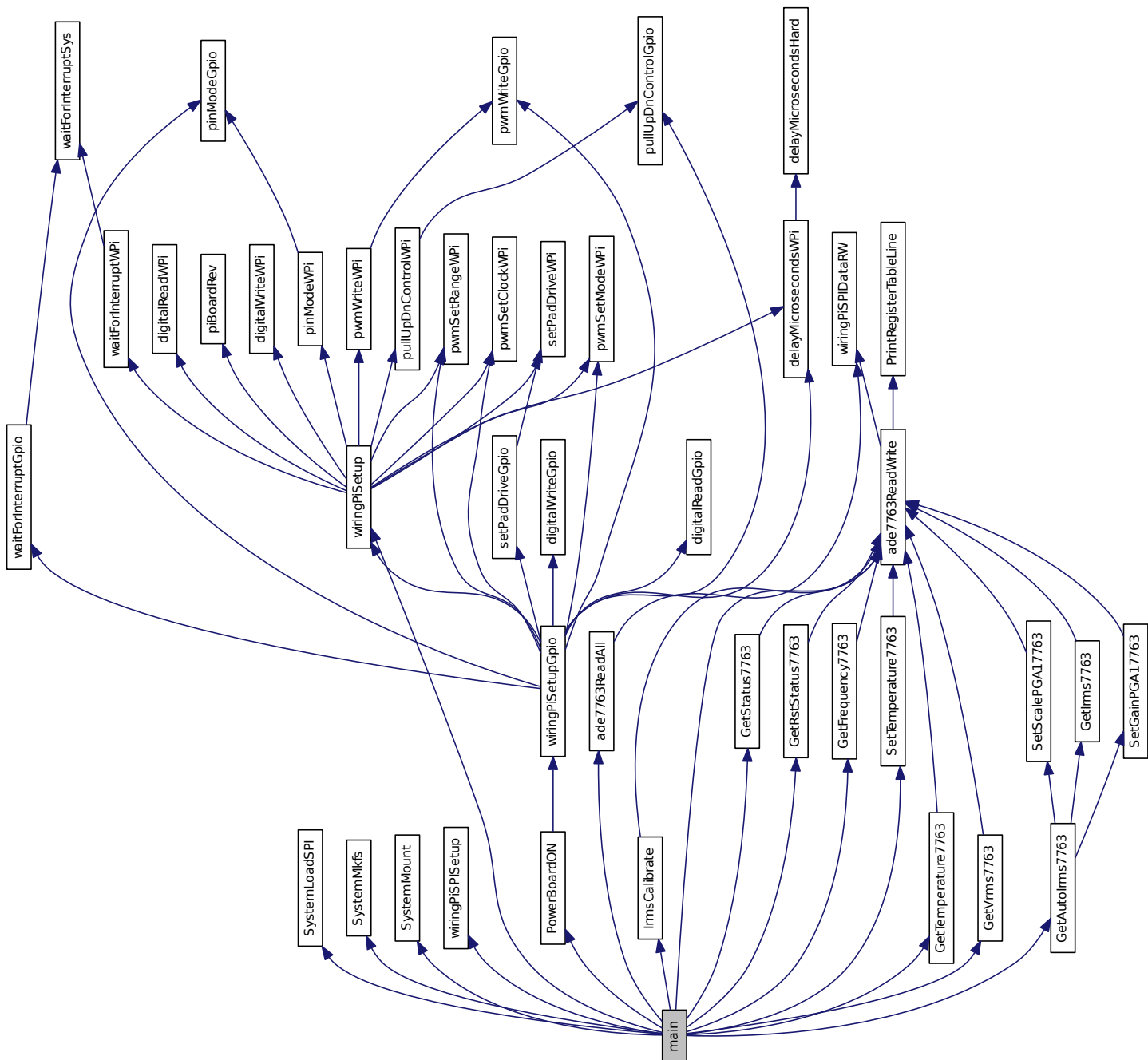
Integrovaný obvod ADE7763 může v „proudové“ měřicí větvi pracovat se sedmi různými citlivostmi vstupních obvodů a třemi různými velikostmi zesílení měřeného signálu. Program musel být vybaven automatickou volbou rozsahu, která nejdříve zjistí (s nejnižší citlivostí a zesílením) přibližný stav odebíraného proudu, respektive velikost napětí na bočníku, a nastaví příslušný měřicí rozsah. Měření je pak provedeno opětovně s vyšší přesností.

---

<sup>1</sup>Výkon je v současné verzi programu dopočítáván jako  $U \times I$ . Nebyly aplikovány pokročilé funkce měření, které poskytuje integrovaný obvod ADE7763.

Integrovaný obvod ADE7763 spouští měření při průchodu síťového napětí nulou. V případě odpojení síťového napětí reaguje obvod na šumové průchody nulou a poskytuje značně zavádějící data. Do řídicího programu byla přidána pojistka pro takovéto stavy a odečet hodnot se provádí pouze v rozmezí frekvencí 45 – 65 Hz.

Samotný kód obsahuje dokumentaci vytvořenou systémem „Doxygen“, která je umístěna ve složce `doc` v hlavním adresáři kódu na přiloženém CD.



Obr. 6.1: Hlavní struktura kódu

## 6.2 Spouštění daemon-a

Aby bylo zařízení schopné plnit funkci automaticky – po startu – bez zásahu technika, musí se binární soubor (měřicí aplikace) spustit na pozadí při startu systému. GNU Linux má pro tento účel spouštěcí skripty umístěné v `/etc/init.d`.

Binární soubor určený pro měření je umístěn v `/usr/bin`. Aby se předešlo potížím s názvy souborů a verzemi, je binárnímu souboru ponechán původní název včetně verze a na tento binární soubor je vytvořen symbolický link, který již neobsahuje verzi sestavení. Tímto postupem je docíleno konstantního nastavení operačního systému a je tak možno jednoduše aktualizovat binární soubory, výpis: 6.1.

**Výpis: 6.1:** Částečný výpis `/usr/bin` – zobrazení umístění měřicího binárního souboru

```
--> ls -la /usr/bin | grep power
lrwxrwxrwx 1 root root      13 kvě  7 21:42 PowerControll -> ./power_1_0_3*
-rwxr-xr-x 1 root root    291K kvě  7 21:40 power_1_0_3*
```

Skript umístěný v `/etc/init.d` může mít následující tvar, výpis: 6.2.

**Výpis: 6.2:** Výpis inicializačního skriptu `power` umístěného v `/etc/init.d`/[4]

```
#!/bin/sh
# /etc/init.d/power
#
#
### BEGIN INIT INFO
# Provides:          PowerControll
# Required-Start:    $sshd
# Required-Stop:     $remote_fs $syslog $sshd
# Should-Start:
# Should-Stop:
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Deamon for communicate width Raspberry power KIT
# Description:        PowerControll is specific user program. It is only
#                      for RaspBerry Power KIT.
#
#
### END INIT INFO

PATH=/bin:/usr/bin:/sbin:/usr/sbin:/usr/local/bin/
# Some things that run always
touch /var/lock/PowerControll

# Carry out specific functions when asked to by the system
case "$1" in
    start)
        echo "Starting PowerControll"
        PowerControll >> /dev/null &
        ;;
    stop)
        echo "Stopping PowerControll"
        killall PowerControll
        ;;
    *)
```



```
echo "Usage: _/etc/init.d/blah_{start|stop}"
exit 1
;;
esac

exit 0
```

Jedaná se o velice jednoduchý skript, který obsahuje několik kritických míst. Prvním kritickým místem je začátek skriptu. Přestože je úvodní část skriptu podle konvence příkazového interpretu **bash** zakomentovaná, nesmí tento blok chybět.

Dalším problematickým místem jsou řádky: **Default-Start** a **Default-Stop** čísla na konci řádku určují, v jakém provozním režimu systému bude aplikace startována a vypínána. Nejdůležitější pro měření je „runlevel 2“ – start do textového režimu a potom „runlevely“ 0 1 6 – jsou vypínací a restartovací režimy. V těchto případech bude běh programu ukončen.

Posledním kritickým místem je řádek začínající: **PATH**. Při startu systém nezná všechny standardní cesty. Proto se mu musí tyto informace tímto řádkem doplnit.

Aby systém zařadil skript do startovací sekvence, je nutné provést následující operaci výpis: 6.3 [4].

**Výpis: 6.3:** Zavedení spouštěcího skriptu do startovací sekvence

```
update-rc.d power defaults # jako root, power je název skriptu
```

**Výpis: 6.4:** Odstranění spouštěcího skriptu ze startovací sekvence

```
update-rc.d -f blah remove
```

## 6.3 Sestavení měření

Měření bylo prováděno po dobu 24 hodin. V tomto časovém úseku měřily současně tři systémy.

1. Monitorování pomocí Raspberry Pi
2. Monitorování pomocí UBiQUiTi mFi
3. Měření pomocí měřicího přístroje METEX

### 6.3.1 Raspberry Pi – podmínky měření

Při tomto měření bylo Raspberry Pi monitorováno v lokální síti pomocí protokolu SNMP. Monitoring prováděl virtuální server s operačním systémem GNU Linux.

Zapojení lokální sítě je vyobrazeno v příloze, obr. B. Data byla ukládána do textového souboru a následně zpracována do grafu pomocí programu Octave. Odečítání se provádělo pravidelně každých 5 s.

### **6.3.2 Systém UBiQUiTi mFi – podmínky měření**

Monitorovací systém UBiQUiTi mFi [24] je podobného charakteru, jakým se zabývá tato práce. Produkt UBiQUiTi mFi je komerčně dostupný produkt určený pro monitorování domácností a menších firemních prostor.

Monitorovací jednotky se připojí k monitorovacímu programu (serveru) po síti LAN. K těmto monitorovacím jednotkám lze připojit dvě čidla. Firma poskytuje v současnosti dveřní senzory, indukční proudové sondy, teplotní senzory a pohybová čidla. V rámci této práce byla využita indukční proudová sonda.

Zařízení bylo spuštěno paralelně s Raspberry Pi, aby bylo možné porovnat naměřená data.

### **6.3.3 METEX – podmínky měření**

Do měřeného proudového okruhu byl zapojen měřicí přístroj METEX M-3850. Pomocí počítače a sběrnice RS-232 byla z měřicího přístroje odečítána naměřená data každých 5 s. Získaná data byla zpracována pomocí Octave do grafu. Data získaná pomocí tohoto přístroje jsou určena pouze k orientačnímu porovnání. Maximální měřená hodnota proudu se pohybovala okolo 1,3 A, což je pro nižší rozsah měřicího přístroje (400 mA) nepřijatelná velikost. Naproti tomu, při měření na rozsahu 20 A, dochází u malých hodnot proudu ke značnému zkreslení vinou zaokrouhlování. Přepínání rozsahu v rámci měření nebylo možné. Přerušením proudového okruhu by došlo k ovlivnění měření Raspberry Pi.

## 6.4 Měření – výsledky

Měření bylo prováděno po dobu jednoho dne od 0:00:00 do 23:59:59. Průběh celého dne, přehled událostí, je zaznamenán v tabulce 6.1.

V kapitole 6.5 jsou zobrazeny a porovnávány grafy, naměřené pomocí tří různých měřicích přístrojů (Raspberry Pi, UBiQUiTi mFi, METEX).

**Tab. 6.1:** Přehled událostí při celodenním měření odběru

Čas	Událost
8:48	Zapnuta pájka – ERS 50 – 60 W
9:03	Zapnuta lampička – klasická žárovka 40 W
9:24	Vypnuta lampička – klasická žárovka 40 W
9:24	Vypnuta pájka – ERS 50 – 60 W
9:56	Zapnuta lampička – klasická žárovka 40 W
10:33	Vypnuta lampička – klasická žárovka 40 W
11:52	Zapnuta spirála na ohřev vody 300 W
12:02	Vypnuta spirála na ohřev vody 300 W
13:04	Zapnuta pájka – ERS 50 – 60 W
13:28	Vypnuta pájka – ERS 50 – 60 W
15:00	Zapnuta lampička – klasická žárovka 40 W
15:35	Vypnuta lampička – klasická žárovka 40 W
17:01	Zapnuta pájka – ERS 50 – 60 W
17:01	Zapnuta lampička – klasická žárovka 40 W
17:26	Vypnuta lampička – klasická žárovka 40 W
17:41	Vypnuta pájka – ERS 50 – 60 W
20:56	Zapnuta lampička – klasická žárovka 40 W
21:51	Vypnuta lampička – klasická žárovka 40 W
21:51	Zapnuta spirála na ohřev vody 300 W
22:04	Vypnuta spirála na ohřev vody 300 W
22:45	Zapnuta nabíječka mobilu – max odběr 0,15 A
23:03	Vypnuta nabíječka mobilu – max odběr 0,15 A
15:00	Zapnuta lampička – klasická žárovka 15 W
15:35	Vypnuta lampička – klasická žárovka 15 W

## 6.5 Porovnání měření Raspberry Pi, UBiQUiTi mFi a METEX

### Měření proudu

Měřicí přístroje METEX a UBiQUiTi mFi jsou schopny měřit pouze proud, nikoliv však výkon. Napětí je při tomto měření konstantní, výkon je tedy přímo úměrný proudu. Porovnávány jsou proto naměřené hodnoty proudu.

Jako první je zobrazen celodenní záznam měření, obr. 6.2, 6.3 a 6.4. Z náhledu je zřejmé, že všechny tři systémy poskytly relativně stejná data<sup>2</sup>.

Z důkladného porovnání naměřených dat vyplývá, že měřicí modul Raspberry Pi lehce podměřuje na nižších hodnotách proudu. Jedná se o konstantní offset na jednotlivých měřicích rozsazích. Tento problém by vyřešila kalibrace přístroje, která však v rámci práce nebyla možná z důvodu nedostupnosti kalibrované zátěže.

Dále je patrná naprostá nemožnost exportu naměřených dat z obslužného programu UBiQUiTi mFi.

### Detailní pohled na významné části proudového grafu

K bližšímu porovnání byla vybrána část grafu, obr. 6.5, 6.6 a 6.7, zobrazující menší proudový odběr (odporová zátěž 40 W). Zde je již dobře patrný záporný offset měřicího modulu Raspberry Pi.

Dalším zkoumaným úsekem je část měření při indukční zátěži (mikropáječka) s častými změnami odebíraného proudu, obr. 6.8, 6.9 a 6.10. I v tomto případě se dá pozorovat pouze offsetový rozdíl naměřených dat.

Poslední zkoumanou částí celodenního měření je úsek grafu zobrazující měření zátěže přibližně 300 W, obr. 6.11, 6.12 a 6.13. Je patrné, že na zvoleném měřicím rozsahu již není offset tak významný, jako u měření předchozích.

Měření zátěže nad 300 W nemohlo být provedeno. Neumožňují to parametry současně osazeného relé.

#### 6.5.1 Měření frekvence

Z grafu, obr. 6.14, je dobře patrné co v napájecí síti způsobí nekvalitní kontakt vypínače. Na počátku měření byla lampa aktivována vypínačem, což způsobilo velké

---

<sup>2</sup>Hodnoty -5 mA v grafu vycházejícím z měření přístrojem METEX signalizují výpadek měření. Data byla přenášena po sběrnici RS232, převodník RS232 – USB v průběhu měření několikrát vysadil. Zejména při změně výkonové zátěže, kdy dochází ke značnému elektromagnetickému rušení. Těmito výpadky mohlo dojít k drobnému časovému posunutí naměřených dat.

rušení v rozvodné síti, ale i mimo ni. Měřicí program v tuto chvíli přerušil měření, jak bylo popsáno v kapitole 6.1, a převodník RS232 – USB ztratil spojení s počítačem.

Díky možnostem měřit frekvenci by zařízení mohlo být využito nejen pro měření spotřeby elektrické energie, ale také pro detekci špatných kontaktů nebo napájecích kabelů měřeného zařízení.

### 6.5.2 Měření napětí

Graf měření napětí, obr. 6.15, nepřináší zásadní informace, ale jsou zde vidět zákmity napájecí sítě při odpojování některých zátěží. Informace o zákmitech napětí mohou sloužit k detekci poruchy v napájecí síti.

### 6.5.3 Měření výkonu

Graf měření výkonu, obr. 6.16, je zcela závislý na již komentovaném a dopodrobna rozebraném grafu odebíraného proudu. Není třeba jej dále komentovat.

### 6.5.4 Chyby měření

#### Chyby měření – METEX

Měřicí přístroj METEX M-3850 prováděl měření proudu na rozsahu 20 A. (Rozsah měření nebylo možné v rámci měření měnit, došlo by k rozpojení proudového okruhu a ovlivnění měření ostatních přístrojů.) Na rozsahu 20 A AC má přístroj definovanou chybu podle rovnice 6.1.

$$\pm 2,0\% \text{ of } rdg + 5 \text{ dgt } (10 \text{ mA}) \quad (6.1)$$

Pomocí matematického programu Octave a rovnice 6.1 byly spočítány chyby měření pro přístroj METEX, tabulka 6.2.

**Tab. 6.2:** Chyby měření METEX

Hodnota	Chyba [mA]
Maximální chyba měření	+27,85 ÷ −27,75
Minimální chyba měření	−0,05 ÷ +0,15
Střední hodnota chyby měření	+1,7735 ÷ −1,6735

## Chyby měření – UBiQUiTi mFi a Raspberry Pi

Chyby měření UBiQUiTi mFi nebylo možné objektivně zhodnotit. Chyby měření Raspberry Pi není možné hodnotit konkrétně. Měřená data neobsahují údaje o zvoleném měřicím rozsahu. Je však možné porovnat absolutní měřenou odchylku od hodnot naměřených přístrojem METEX.

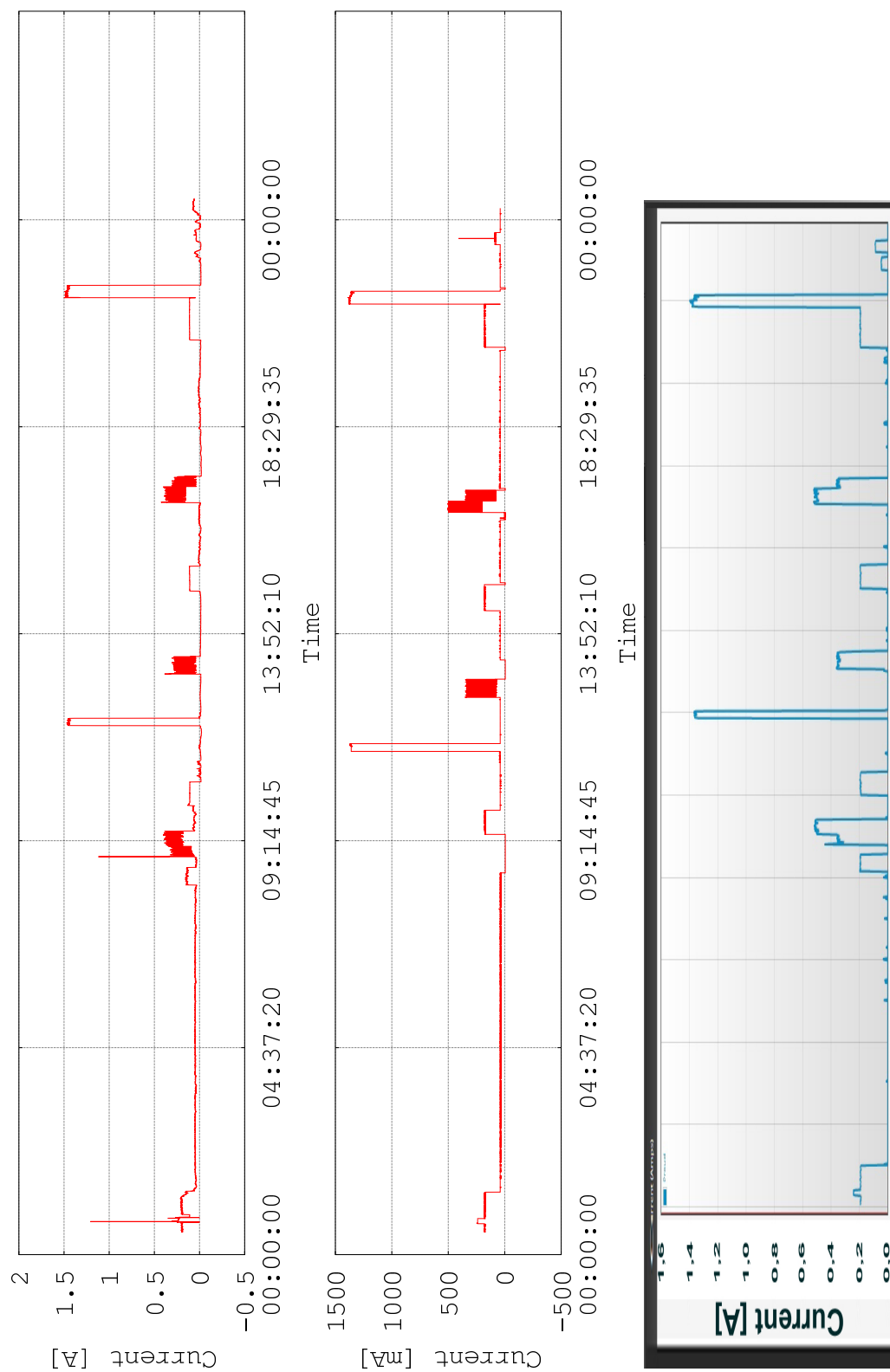
Měřicí přístroje prováděly odečty hodnot za stejný časový úsek, nikoliv však ve stejný časový okamžik. Tato skutečnost značně komplikuje matematické porovnání dat, získaných přístrojem METEX a systémem Raspberry Pi. Pro získání absolutní odchylky hodnot poskytnutých Raspberry Pi od hodnot měřených přístrojem METEX byly hodnoty Raspberry Pi nejdříve odečteny od hodnot přístroje METEX. Následně byla odstraněna data, která díky časovému nesouladu nebyla postižena rozdílůváním hodnot. Na závěr byly získané hodnoty podrobeny filtrování plovoucím, průměrovacím oknem o délce devět prvků.

Výsledkem těchto operací jsou dva grafy, obr. 6.17 a 6.18. Graf, obr. 6.17, zobrazuje absolutní odchylky měření Raspberry Pi od hodnot měřených přístrojem METEX. Druhý graf, obr. 6.18, zobrazuje tutéž situaci. Absolutní odchylka (její amplituda) je zde vynásobena pěti, pro lepší orientaci v grafu. V obou grafech jsou pro snadnější orientaci ponechány hodnoty měřeného proudu přístrojem METEX.

Z grafu, obr. 6.18, je patrná závislost absolutní odchylky měřeného proudu na velikosti měřených veličin, potažmo na zvoleném měřicím rozsahu. Maximální odchylka hodnot, měřených pomocí Raspberry Pi, od hodnot, naměřených pomocí přístroje METEX, byla určena na 72,4 mA. Průměrná odchylka se pohybovala okolo 22.7 mA.

## 6.6 Vzdálené měření

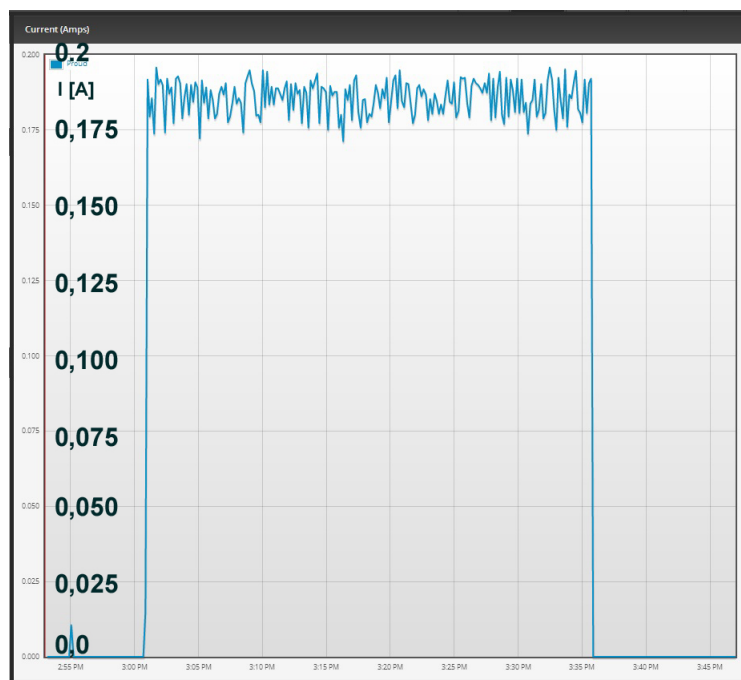
Hlavní myšlenkou této práce bylo měřit spotřebu elektrické energie na místě uživateli vzdáleném. Testování těchto vlastností probíhalo za technické podpory společnosti „Hanácké.net“. Zařízení bylo zapojeno do počítačové sítě, obr. B. Výchozí brána této lokální sítě, umístěna ve Vídni a připojena k internetu pomocí PPPoE, vytvořila jeden přímý VPN tunel do počítačové sítě v Olomouci, kde byl umístěn monitorovací server se systémem Zabbix, a dále jeden záložní VPN tunel do Prahy (do sítě stejné společnosti). Na monitorovacím serveru bylo nastaveno sledování Raspberry Pi. Výsledné grafy jsou umístěny v příloze, obr. C.1, C.2, C.3 a C.4.



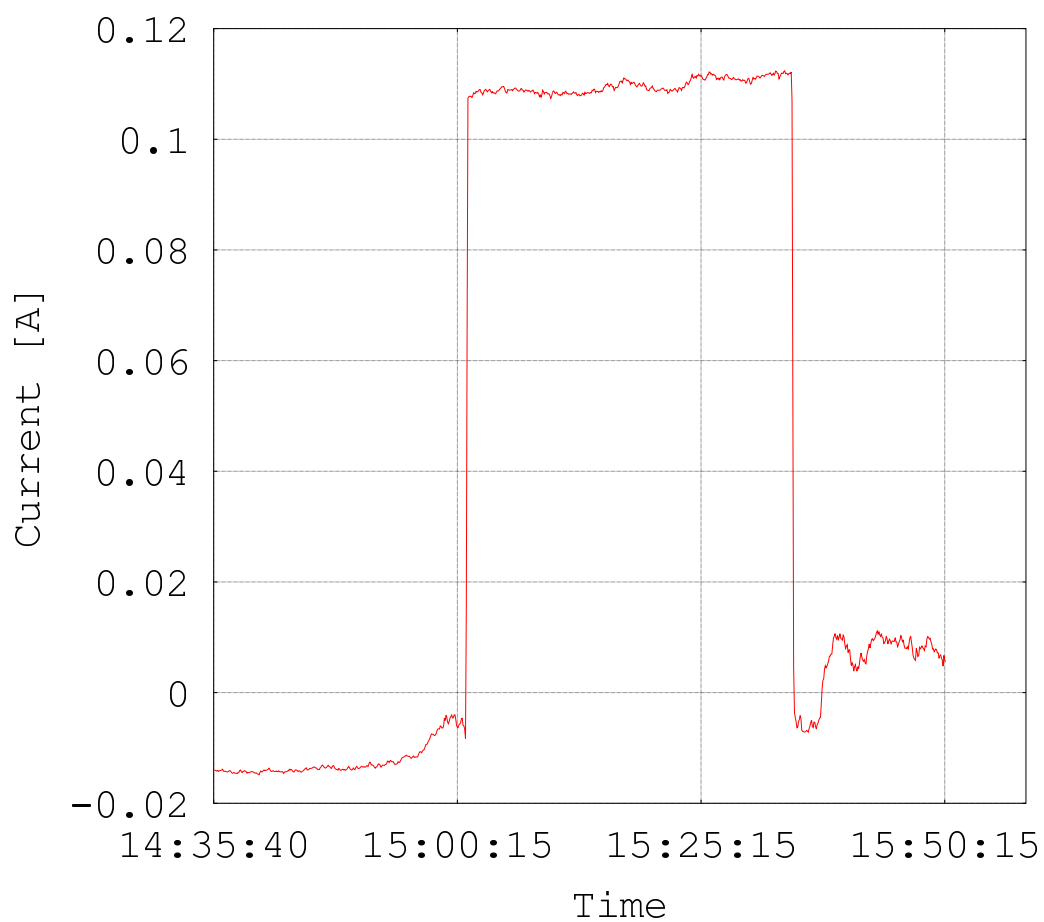
**Obr. 6.2:** Průběh odebíraného proudu za 24 hod. – měřeno pomocí Raspberry Pi

**Obr. 6.3:** Průběh odebíraného proudu za 24 hod. – měřeno pomocí METEX

**Obr. 6.4:** Průběh odebíraného proudu za 24 hod. – měřeno pomocí UBiQUiTi mFi

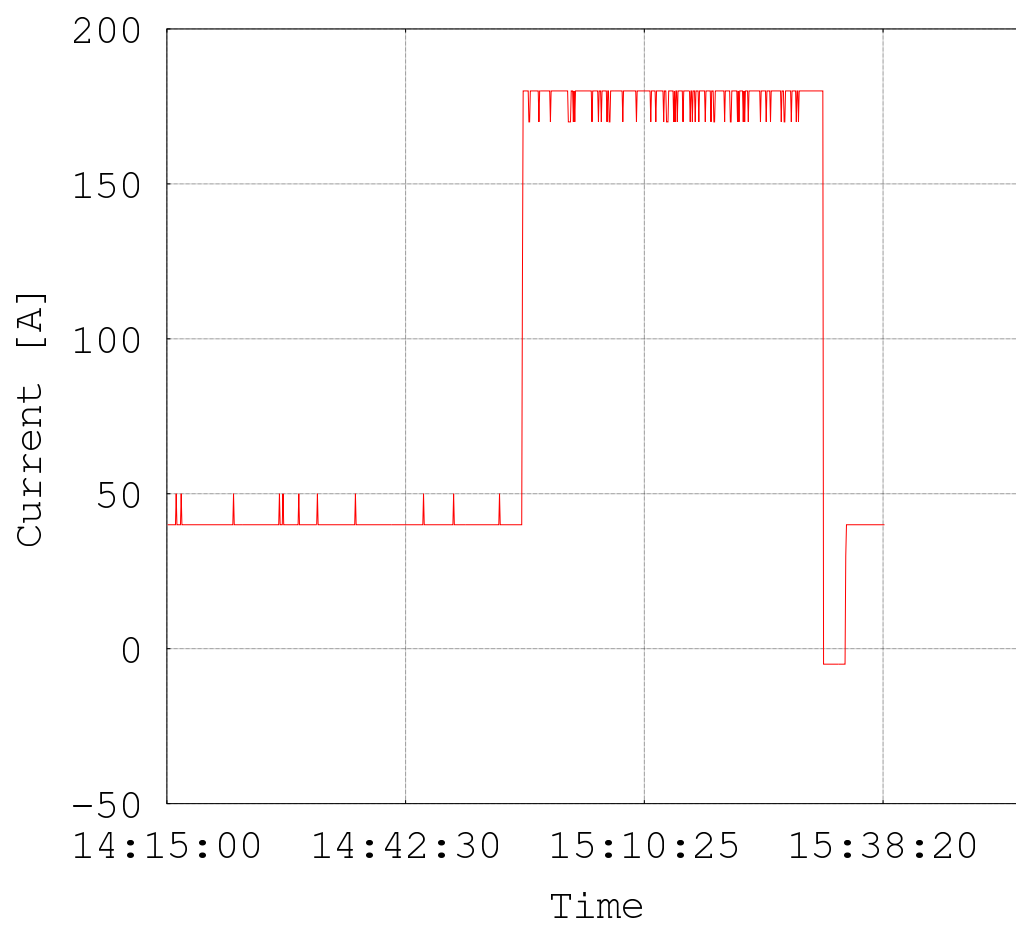


**Obr. 6.5:** Průběh odebíraného proudu zoom 15:00 – 15:35 hod. – měřeno pomocí UBiQUiTi mFi

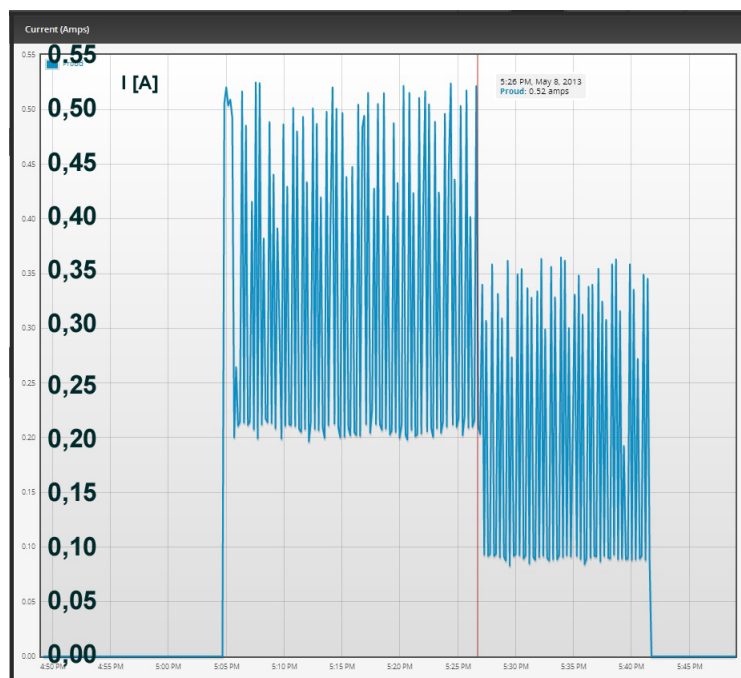


**Obr. 6.6:** Průběh odebíraného proudu zoom 15:00 – 15:35 hod. – měřeno pomocí Raspberry Pi

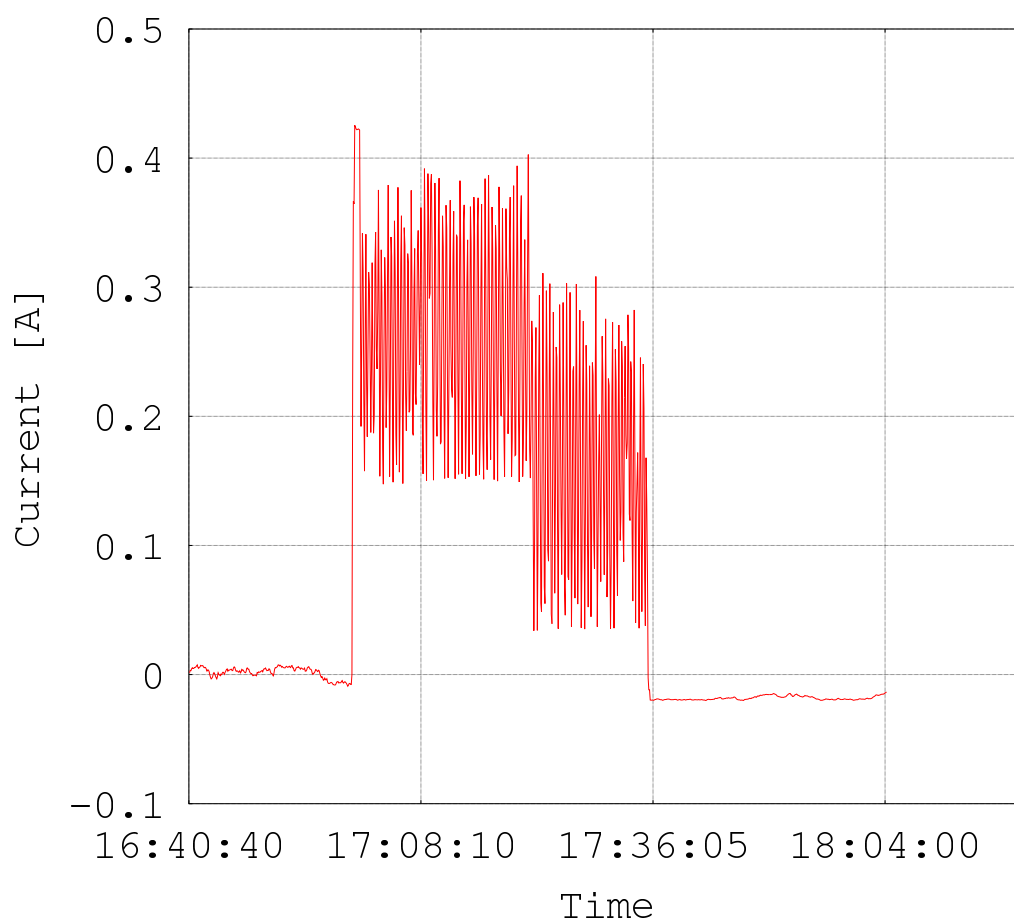




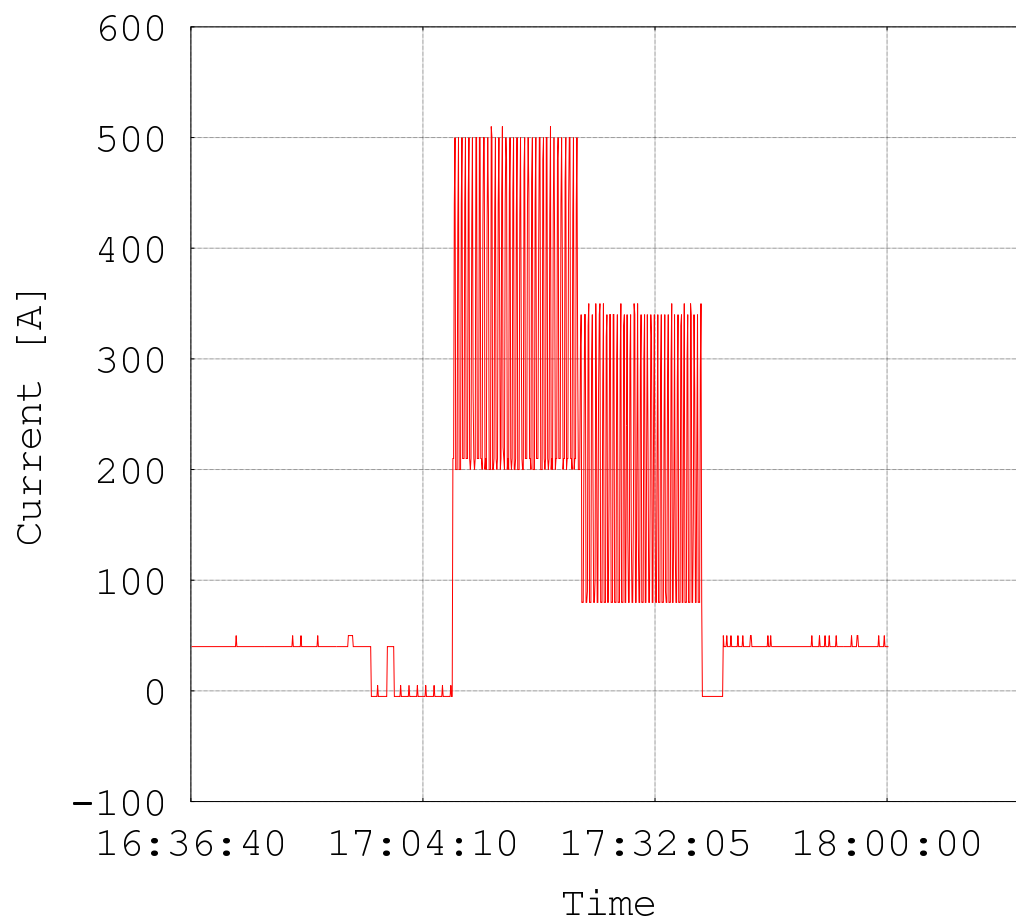
**Obr. 6.7:** Průběh odebíraného proudu zoom 15:00 – 15:35 hod. – měřeno pomocí METEX



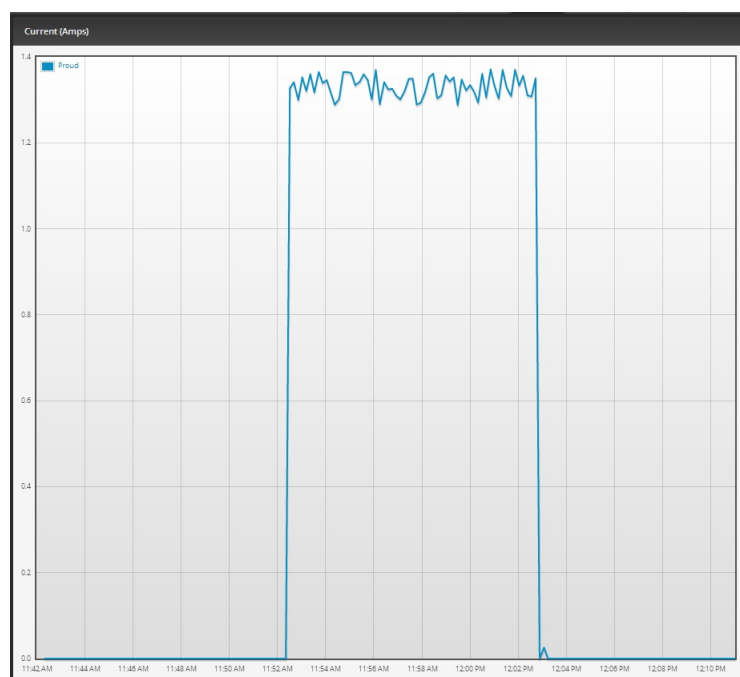
**Obr. 6.8:** Průběh odebíraného proudu zoom 17:01 – 17:41 hod. – měřeno pomocí UBiQUiTi mFi



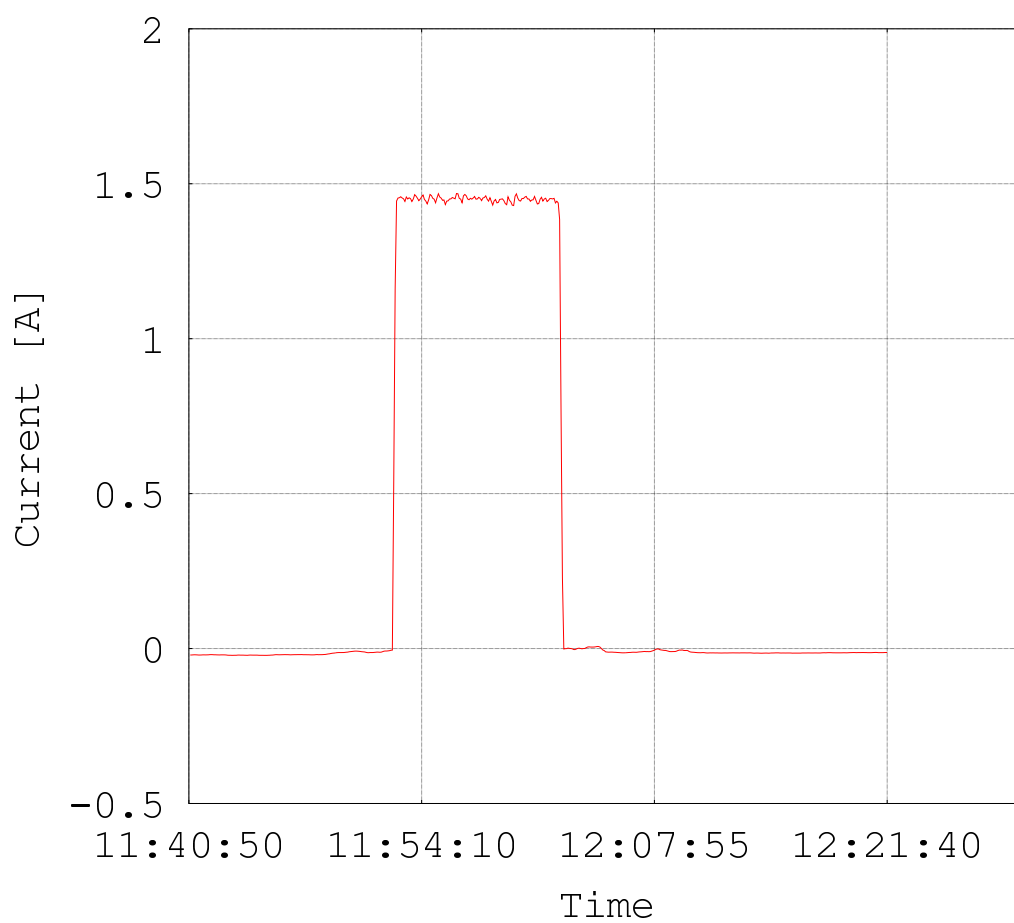
**Obr. 6.9:** Průběh odebíraného proudu zoom 17:01 – 17:41 hod. – měřeno pomocí Raspberry Pi



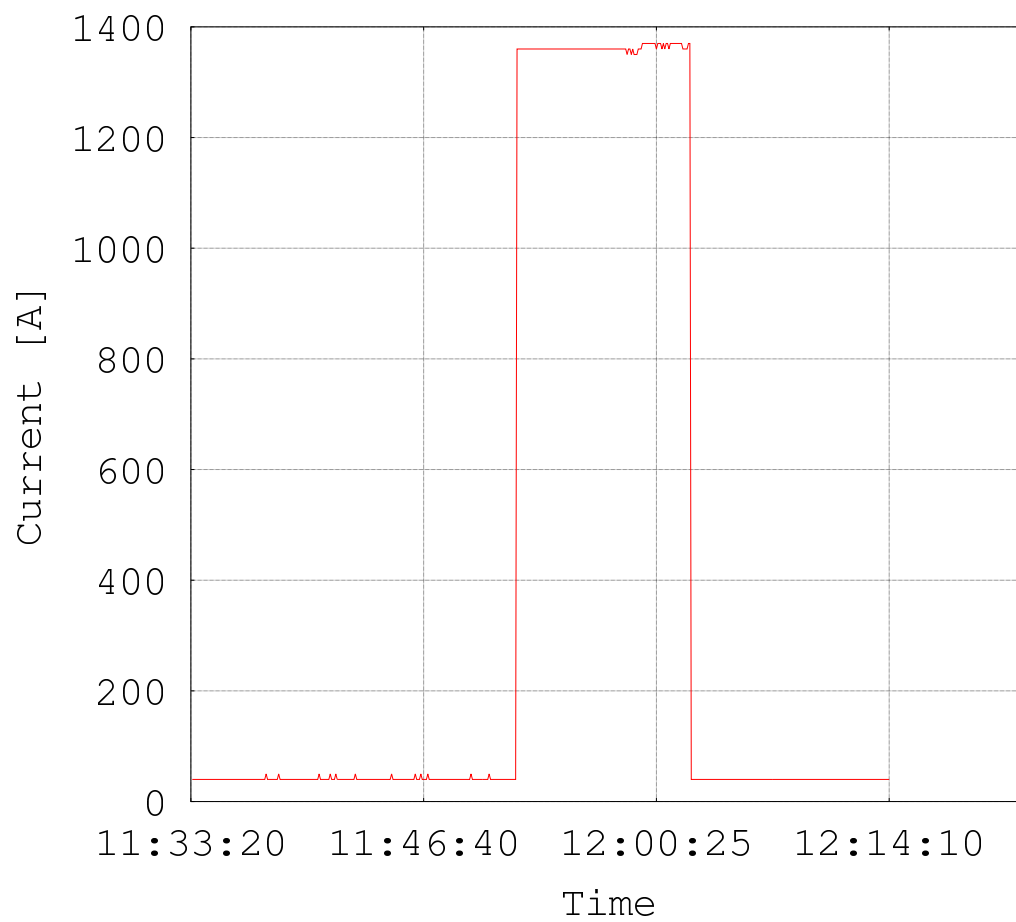
**Obr. 6.10:** Průběh odebíraného proudu zoom 17:01 – 17:41 hod. – měřeno pomocí ME-TEX



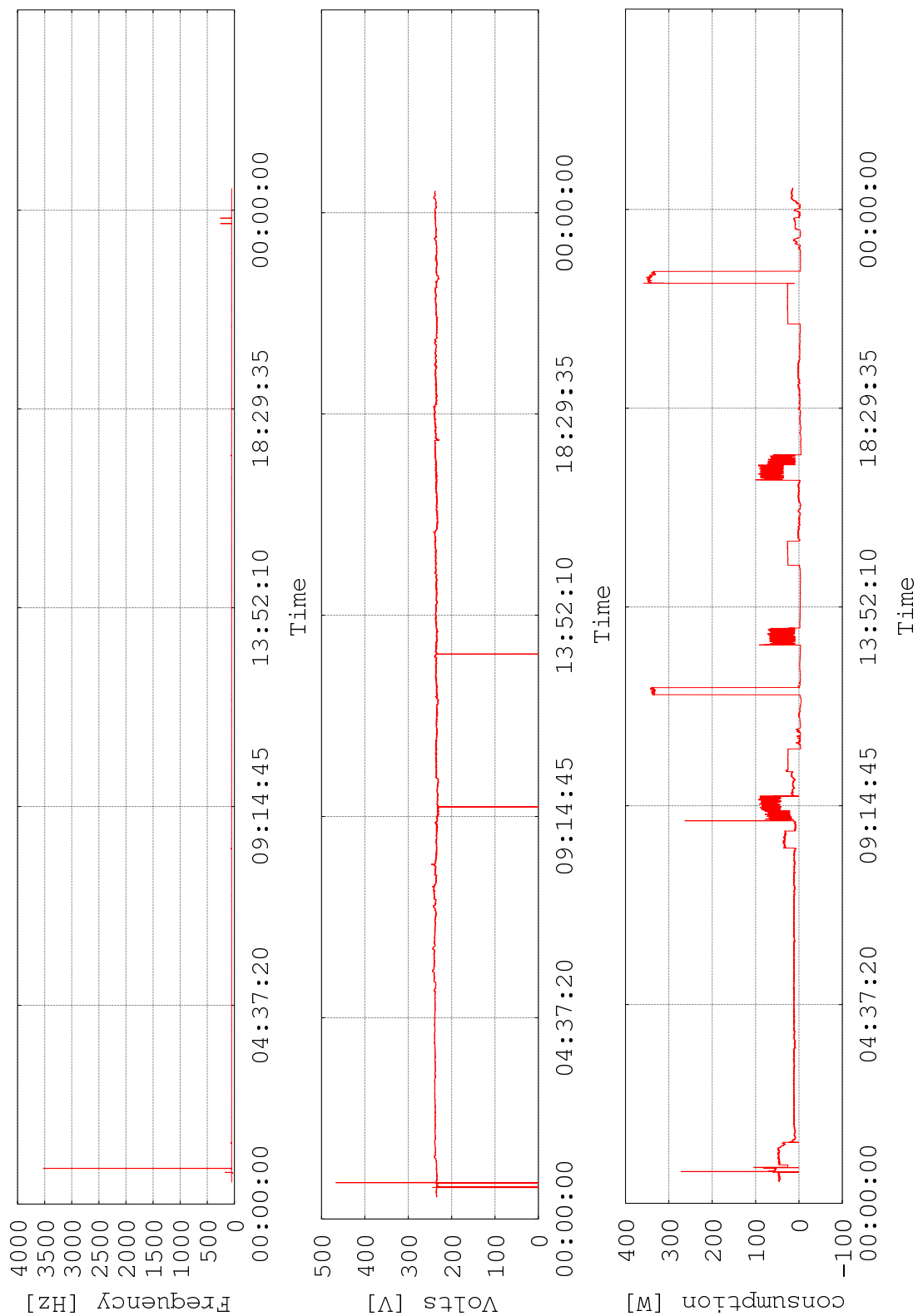
**Obr. 6.11:** Průběh odebíraného proudu zoom 11:52 – 12:02 hod. – měřeno pomocí UBiQUiTi mFi



**Obr. 6.12:** Průběh odebíraného proudu zoom 11:52 – 12:02 hod. – měřeno pomocí Raspberry Pi



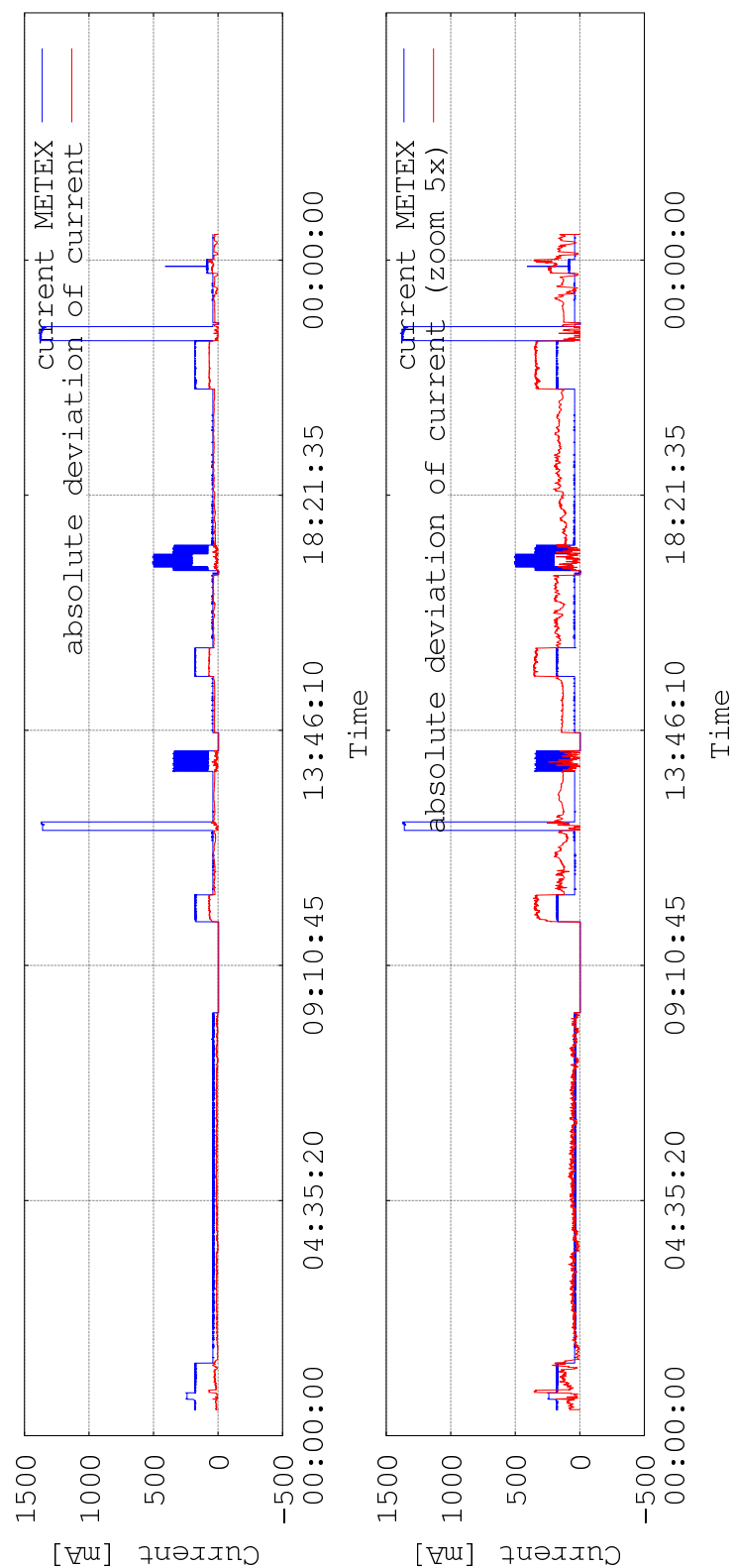
**Obr. 6.13:** Průběh odebíraného proudu zoom 11:52 – 12:02 hod. – měřeno pomocí ME-TEX



**Obr. 6.14:** Průběh síťové frekvence za 24 hod. – měřeno pomocí Raspberry Pi

**Obr. 6.15:** Průběh síťového napětí za 24 hod. – měřeno pomocí Raspberry Pi

**Obr. 6.16:** Průběh odebíraného výkonu za 24 hod. – měřeno pomocí UBiQuiTi mFi



**Obr. 6.17:** Absolutní odchylka měření Raspberry Pi od měření přístrojem METEX, v porovnání s měřenou hodnotou proudu přístrojem METEX

**Obr. 6.18:** Absolutní odchylka měření Raspberry Pi od měření přístrojem METEX (amplituda vynásobena 5), v porovnání s měřenou hodnotou proudu přístrojem METEX

## 7 POROVNÁNÍ VLASTNOSTÍ SYSTÉMU RASPBERRY PI SE SYSTÉMEM UBIQUITI MFI

Systém UBiQUiTi mFi je postavený na monitorovacím programu sdružujícím univerzální sběrné jednotky (připojené po síti LAN), ke kterým je možno připojit několik typů čidel. Celý systém vypadá velice efektně. Uživatel si v dohledovém centru může vymodelovat 3D model např. svého bytu. Do tohoto modelu lze graficky umístit jednotlivá čidla, u kterých jsou následně k dispozici kvalitně zpracované grafy měření. Tento systém má ale značné nedostatky v manipulaci s naměřenými daty – nebyla zde nalezena možnost data exportovat. Další zvláštnost systému spočívá v samotné instalaci. Do počítače je instalován monitorovací program, který při inicializaci hledá dostupné sběrné jednotky. S těmito jednotkami se spáruje a zamezí tak přístup na směrná místa z jiného počítače. Data jsou sice dostupná přes webové rozhraní, ale reinstalace monitorovacího programu nebo jeho instalace na jiný počítač vede k obnovení výchozího nastavení všech použitých jednotek.

Měřicí jednotka postavená na Raspberry Pi sdružuje měřicí modul a sběrné místo do jednoho prvku. Na konci této práce je k dispozici pouze modul pro měření spotřeby elektrické energie. Modularita celého systému však počítá s možností dalšího rozšiřování. Grafická prezentace dat je u tohoto systému možná dvěma způsoby.

V rámci samotného zařízení jde pouze o jednoduchou webovou stránku poskytující aktuální data. K datům je tak možné přistupovat neprodleně po aktivaci zařízení pomocí webového prohlížeče, bez ohledu na software instalovaný na daném počítači.

Druhý způsob prezentace dat záleží čistě na uživateli. Zařízení (Raspberry Pi) poskytne naměřená data jakémukoliv monitorovacímu systému, který je schopen SNMP data číst, a zároveň má povolený přístup k zařízení. Problém zabezpečení také řeší protokol SNMP. Data je tak možné pozorovat od přehledu v textovém souboru až po složité grafy v monitorovacích systémech. Záleží pouze na uživateli, jakou možnost zvolí. Značná většina systémů data ukládá do databáze na serverech, které jsou automaticky zálohovány, čímž je vyřešeno i zálohování získaných dat.



## 8 ZÁVĚR

Práce se zabývá několika tématickými okruhy. Postupně je podán detailní popis problematiky zařízení pro vzdálené monitorování spotřeby elektrické energie.

První kapitola mluví obecně o přístupu k měření elektrických veličin. Výsledkem této části je výběr obvodu ADE7763 určeného pro měření spotřeby elektrické energie. Tento obvod je klíčovou částí celého projektu.

Do úvodní části práce patří kapitola hovořící o možnostech přenosu naměřených dat po internetu nebo v rozlehlé síti. Je zde popsán protokol SNMP, jenž je přímo určen k získávání dat a stavových informací síťových prvků a jiných zařízení monitorovaných po síti. V této kapitole jsou popsány také dva monitorovací systémy, určené pro sběr dat pomocí SNMP protokolu a následné grafické zpracování těchto získaných dat.

Praktickou část práce zahajuje kapitola zabývající se návrhem zařízení pro měření spotřeby elektrické energie. Samotné zařízení je rozděleno na dvě části. První část je určena pro řízení celé aplikace a pro komunikaci zařízení s vnější sítí. Za tímto účelem byl vybrán mini-počítač Raspberry Pi s nasazeným operačním systémem Raspbian. Druhou část tvoří navržené moduly pro samotné měření elektrických veličin. V rámci práce byly zpracovány dvě verze měřicích modulů. Obě byly osazeny integrovaným obvodem ADE7763, galvanickým oddělením měřené části (část obvodu pod napětím 230 V) od části komunikující s Raspberry Pi, a dále bistabilním relé pro možnost odpojit měřený okruh od síťového napětí. Druhá verze byla více přizpůsobena rozměrům Raspberry Pi a byla zde zdokonalena ochrana Raspberry Pi před vniknutím síťového napětí na desku Raspberry Pi.

Práce se dále podrobně zabývá nastavením protokolu SNMP pro operační systém Raspbian. Je zde popsána konfigurace většiny možností protokolu SNMP. Variant nastavení je nespočetně a použitá konfigurace se může lišit od vzorového nastavení. Přístup k problematice nastavení je však stejný. Následně byl okrajově zmíněn provoz webového serveru, jenž doplňuje široké možnosti zařízení, a také možnost vzdálené správy zařízení pomocí protokolu SSH.

Poslední kapitoly práce se zabývají nejprve zatížením přenosové sítě, monitorovaného a monitorujícího zařízení v závislosti na počtu přenášených dat, následně pak samotným měřením elektrických veličin.

Bylo ověřeno, že přenos dat nijak významně nezatěžuje přenosovou síť a ani samotné čtení dat klientem (monitorovacím serverem) a poskytování dat agentem běh operačního systému příliš nezatíží. Díky tomuto zjištění bylo rozhodnuto, že samotné zařízení bude uchovávat pouze aktuální data v paměti RAM, aby nedocházelo ke zbytečnému opotřebení paměťové SD karty.

Práci uzavírá kapitola měření. V této části práce byl napsán obslužný program

v jazyce C. Tento program komunikuje po sběrnici SPI s obvodem ADE7763 a odečítá aktuální data napětí, proudu, frekvence a dopočítává hodnoty výkonu. Je možné jej dále rozšířit na přímé odečítání výkonu z integrovaného obvodu ADE7763. Odečítání výkonu potřebuje přesnou kalibraci měřených veličin napětí a proudu. To však z důvodů absence<sup>1</sup> potřebných zařízení nebylo v rámci této práce realizováno. Do obslužného programu byly implementovány funkce pro průměrování naměřených dat a volba automatického rozsahu měřeného proudu.

Samotné měření bylo prováděno třemi nezávislými systémy (METEX, UBiQUiTi mFi a Raspberry Pi) a probíhalo po dobu jednoho dne. Z naměřených dat bylo možné posoudit vlastnosti navrženého systému. Naměřená data pomocí systému postaveného na Raspberry Pi odpovídají datům získaným ze zbylých dvou měřicích přístrojů. Jediné zaznamenané odchylky byly pozorovány v konstantním offsetu měřeného proudu. Tento problém by bylo možné odstranit pomocí přesné kalibrace zařízení, jak bylo zmíněno v předchozím odstavci. Ale i bez této kalibrace jsou získaná data použitelná k orientačnímu měření spotřeby elektrické energie. Zařízení poskytlo velice kvalitní data i z ostatních měření (napětí a frekvence). Díky těmto výsledkům lze posoudit kvalitu napájecí sítě a popřípadě odhalit problémy v rozvodné síti.

Zařízení bylo v rámci testování připojeno na vzdálený monitorovací server (Vídeň – Olomouc). Spojení probíhalo za naprosto reálných podmínek internetového provozu. Vzdálený server úspěšně zaznamenal, pomocí monitorovacího systému Zabbix, data naměřená pomocí Raspberry Pi.

V úvodu práce byl stanoven cíl: navrhnout kompaktní zařízení, schopné monitorovat spotřebu elektrické energie jakéhokoli i vzdáleného zařízení a poskytnout naměřená data uživateli prostřednictvím počítačové sítě ve standardizované podobě. Při závěrečném hodnocení lze konstatovat úspěšné splnění všech kladených cílů.

---

<sup>1</sup>Tato diplomová práce byla dokončována mimo VUT Brno (na zahraniční stáži).

## LITERATURA

- [1] ANALOG DEVICES, Inc. *ADE7755 (Rev.A): Energy Metering IC with Pulse Output* [online]. Rev.A. Norwood, U.S.A., 2009 [cit. 2012-04-28]. ISBN D02896-0-8/09(A). Dostupné z: <[http://www.analog.com/static/imported-files/data\\_sheets/ADE7755.pdf](http://www.analog.com/static/imported-files/data_sheets/ADE7755.pdf)>.
- [2] ANALOG DEVICES, Inc. *Single-Phase Active and Apparent Energy Metering IC: ADE7763* [online]. Rev. B. 2009, 9. červen 2011 [cit. 2012-12-09]. ISBN -. Dostupné z: <[http://www.analog.com/static/imported-files/data\\_sheets/ADE7763.pdf](http://www.analog.com/static/imported-files/data_sheets/ADE7763.pdf)>.
- [3] BROADCOM EUROPE LTD. *BCM2835 ARM Peripherals* [online]. 406 Science Park Milton Road Cambridge CB4 0WW, 2012, 6. 2. 2012 [cit. 2012-12-09]. ISBN -. Dostupné z: <<http://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>>.
- [4] Debian Administration: Making scripts run at boot time with Debian. Debian Administration [online]. 2004 [cit. 2013-05-11]. Dostupné z: <<http://www.debian-administration.org/articles/28>>.
- [5] D'SOUZA. MICROCHIP TECHNOLOGY INC. *TB008: Transformerless Power Supply* [online]. USA, 2000 [cit. 2012-04-28]. ISBN DS91008B. Dostupné z: <<http://www.jimfranklin.info/microchipdatasheets/91008b.pdf>>.
- [6] ELEMENT14. *Quick Start Guide: The Raspberry Pi – Single Board Computer* [online]. 2012 [cit. 2012-12-09]. ISBN -. Dostupné z: <[www.farnell.com/datasheets/1524403.pdf](http://www.farnell.com/datasheets/1524403.pdf)>.
- [7] EMBEDDED LINUX WIKI. *RPi Projects* [online]. 2012, 2.12. 2012 [cit. 2012-12-09]. Dostupné z: <[http://elinux.org/RPi\\_Projects](http://elinux.org/RPi_Projects)>.
- [8] Energy Measurement. MICROCHIP TECHNOLOGY INC. *Microchip* [online]. 2012 [cit. 2012-04-28]. Dostupné z: <<http://www.microchip.com/ParamChartSearch/chart.aspx?branchID=11029&mid=11&lang=en&pageId=79>>.
- [9] Fedora Documentation: Monitoring Performance with Net-SNMP. FEDORA. *Fedora Documentation* [online]. 2012 [cit. 2012-12-01]. Dostupné z: <[http://docs.fedoraproject.org/en-US/Fedora/16/html/System\\_Administrators\\_Guide/sect-System\\_Monitoring\\_Tools-Net-SNMP.html](http://docs.fedoraproject.org/en-US/Fedora/16/html/System_Administrators_Guide/sect-System_Monitoring_Tools-Net-SNMP.html)>.

- [10] ICL7107 / ICL7106 - DIGITAL VOLTMETER. ELECTRONICS-DIY.COM. *Electronics-DIY* [online]. 2002-2012 [cit. 2012-04-28]. Dostupné z: <[http://electronics-diy.com/ICL7107\\_volt\\_meter.php](http://electronics-diy.com/ICL7107_volt_meter.php)>.
- [11] J. CASE, M. FEDOR, M. SCHOFFSTALL a J. DAVIN. RFC:1157. *A Simple Network Management Protocol: (SNMP)*. Organization for Standardization, International Standard: MIT Laboratory for Computer Science, May 1990. Dostupné z: <<http://www.ietf.org/rfc/rfc1157.txt>>.
- [12] MRÁZEK, Oldřich. SNMP protokol a jeho využití. HW SERVER S.R.O. *Hw.cz* [online]. 2003 [cit. 2012-04-29]. Dostupné z: <<http://www.hw.cz/produkty/snmpprotokol-a-jeho-vyuziti.html>>.
- [13] OBSERVIVUM. OBSERVIVUM: network management and monitoring [online]. 2013 [cit. 2013-05-11]. Dostupné z: <[http://www.observivum.org/wiki/Main\\_Page](http://www.observivum.org/wiki/Main_Page)>.
- [14] Open Source @ seneca: Raspberry Pi Fedora Remix. Open Source @ seneca [online]. 2013 [cit. 2013-05-10]. <Dostupné z: [http://zenit.senecac.on.ca/wiki/index.php/Raspberry\\_Pi\\_Fedora\\_Remix](http://zenit.senecac.on.ca/wiki/index.php/Raspberry_Pi_Fedora_Remix)>.
- [15] O'REILLY: The 20-Minute SNMP Tutorial - Automating System Administration with Perl. O'REILLY: Spreading the knowledge of innovators [online]. 2013 [cit. 2013-05-10]. Dostupné z: <<http://oreilly.com/perl/excerpts/system-admin-with-perl/twenty-minute-snmpp-tutorial.html>>.
- [16] Part Number Search :STPM. STMICROELECTRONICS. *STMicroelectronics* [online]. 2011 [cit. 2012-04-28]. Dostupné z: <<http://www.st.com/stonline/stappl/productcatalog/app?page=partNumberSearchPage&search=STPM>>.
- [17] Ramdisk: Paměť místo disku. MARTIN ŠÍŇ. *Linuxexpres.cz* [online]. 2009, Čtvrtek, 10. září [cit. 2012-12-09]. Dostupné z: <<http://www.linuxexpres.cz/praxe/ramdisk-pamet-misto-disku>>.
- [18] RDONS PROJECTS. *WiringPi* [online]. 2012 [cit. 2012-12-13]. Dostupné z: <<https://projects.drogon.net/raspberry-pi/wiringpi/>>.
- [19] Sames - All Products. SAMES. *SAMES: Total Solutions in Silicon* [online]. 2012 [cit. 2012-04-28]. Dostupné z: <[http://www.sames.co.za/index.php?option=com\\_content&task=view&id=40&Itemid=55](http://www.sames.co.za/index.php?option=com_content&task=view&id=40&Itemid=55)>.

- [20] SANDISK CORPORATION. *SanDisk SD Card: Product Manual* [online]. Version 2.2. 2004 [cit. 2012-12-08]. Document No. 80-11-00160. Dostupné z: <[http://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&sqi=2&ved=0CC4QFjAA&url=http%3A%2F%2Fwww.flashgenie.net%2Fimg%2Fproductmanualsdcardv2.2final.pdf&ei=xrHDUL2ZI4iWswa2h4D4CQ&usg=AFQjCNFiN9aKLFb2oULMtxLaP9Q4dzSwiw&sig2=-X0\\_ZGyuCuaB00xC4EQW1Q](http://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&sqi=2&ved=0CC4QFjAA&url=http%3A%2F%2Fwww.flashgenie.net%2Fimg%2Fproductmanualsdcardv2.2final.pdf&ei=xrHDUL2ZI4iWswa2h4D4CQ&usg=AFQjCNFiN9aKLFb2oULMtxLaP9Q4dzSwiw&sig2=-X0_ZGyuCuaB00xC4EQW1Q)>.
- [21] Selection Table for Single Phase Metering. ANALOG DEVICES, Inc. *Analog Devices* [online]. 2012 [cit. 2012-04-28]. Dostupné z: <<http://www.analog.com/ps/psthandler.aspx?pstid=10210&la=en#>>.
- [22] SEPPÄNEN, Samuli. SNMP. DEBIAN. *Debian wiki* [online]. 2012, 2012-09-28 17:28:31 [cit. 2012-12-01]. Dostupné z: <<http://wiki.debian.org/SNMP>>.
- [23] Solarwinds. IpMonitor Support Portal: SNMP Center ħ Global Oid Tree [online]. 2013 [cit. 2013-05-15]. Dostupné z: <[http://support.ipmonitor.com/mibs\\_byoidtree.aspx](http://support.ipmonitor.com/mibs_byoidtree.aspx)>.
- [24] UBIQUITI NETWORKS. Ubiquiti networks: mFi [online]. 2013 [cit. 2013-05-11]. Dostupné z: <<http://www.ubnt.com/mfi>>.
- [25] Ubuntu.cz: Apache s MySQL a PHP. UBUNTU. Ubuntu.cz: Ubuntu Wiki [online]. 2012 [cit. 2013-05-11]. Dostupné z: <[http://wiki.ubuntu.cz/server/apache\\_s\\_mysql\\_a\\_php?s\[\]=lamp](http://wiki.ubuntu.cz/server//apache_s_mysql_a_php?s[]=lamp)>.
- [26] ZABBIX SIA. Zabbix: The Enterprise-class Monitoring Solution for Everyone [online]. 2013 [cit. 2013-05-11]. Dostupné z: <<http://www.zabbix.com/>>.

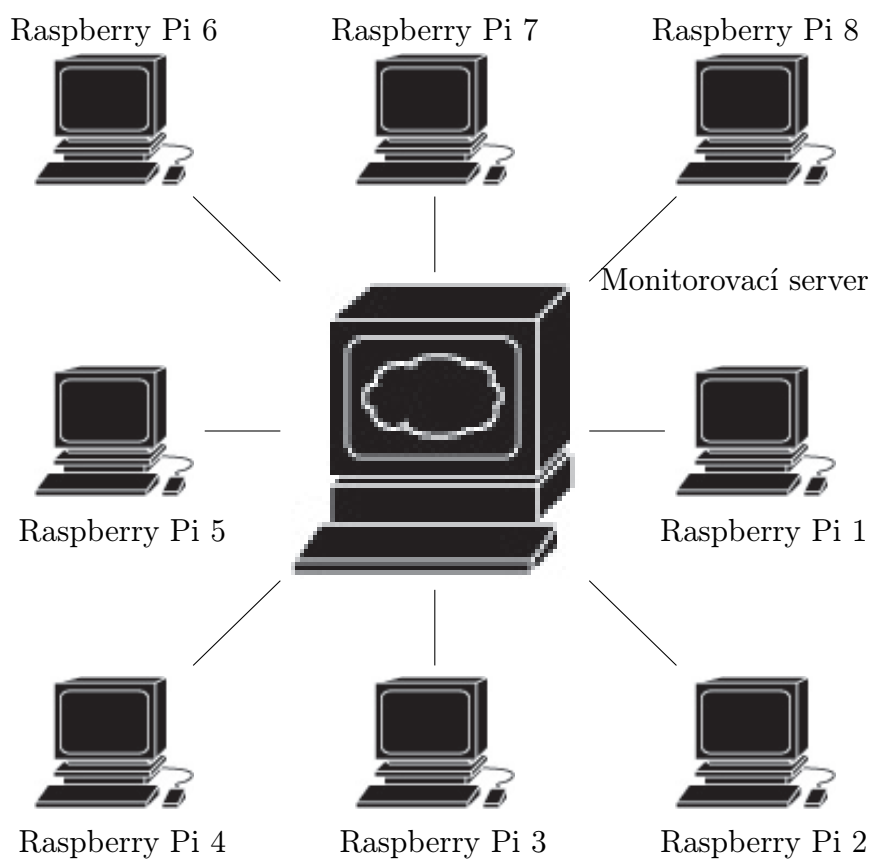
## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

A/D	převodník Analog – Digital
CS	Chip Select – výběr aktuálně komunikujícího prvku
DPS	Deska Plošných Spojů
GNU Linux	GNU's Not Unix, Linux – operační systém distribuovaný pod licenci GPL
GPIO	General Purpose Input/Output
GPL	General Public License
HW	Hardware – fyzický
I <sup>2</sup> C	Inter-Integrated Circuit [I-squared-C]
LAN	Local Area Network
LCD	Liquid crystal display
MIB	Management Information Base
OS	Operační Systém
POE	Power Over Ethernet
PPPoE	Point-to-point protocol over Ethernet
QPL-1.0	The Q Public License Version
RAM	random-access memory
SD	Secure Digital – typ paměťové karty
SNMP	Simple Network Management Protocol
snmpd	Simple Network Management Protocol Daemon
SPI	Serial Peripheral Interface
SSH	Secure Shell
UART	Universal Asynchronous Receiver and Transmitter
USB	Universal Serial Bus
VPN	Virtual Private Network

## SEZNAM PŘÍLOH

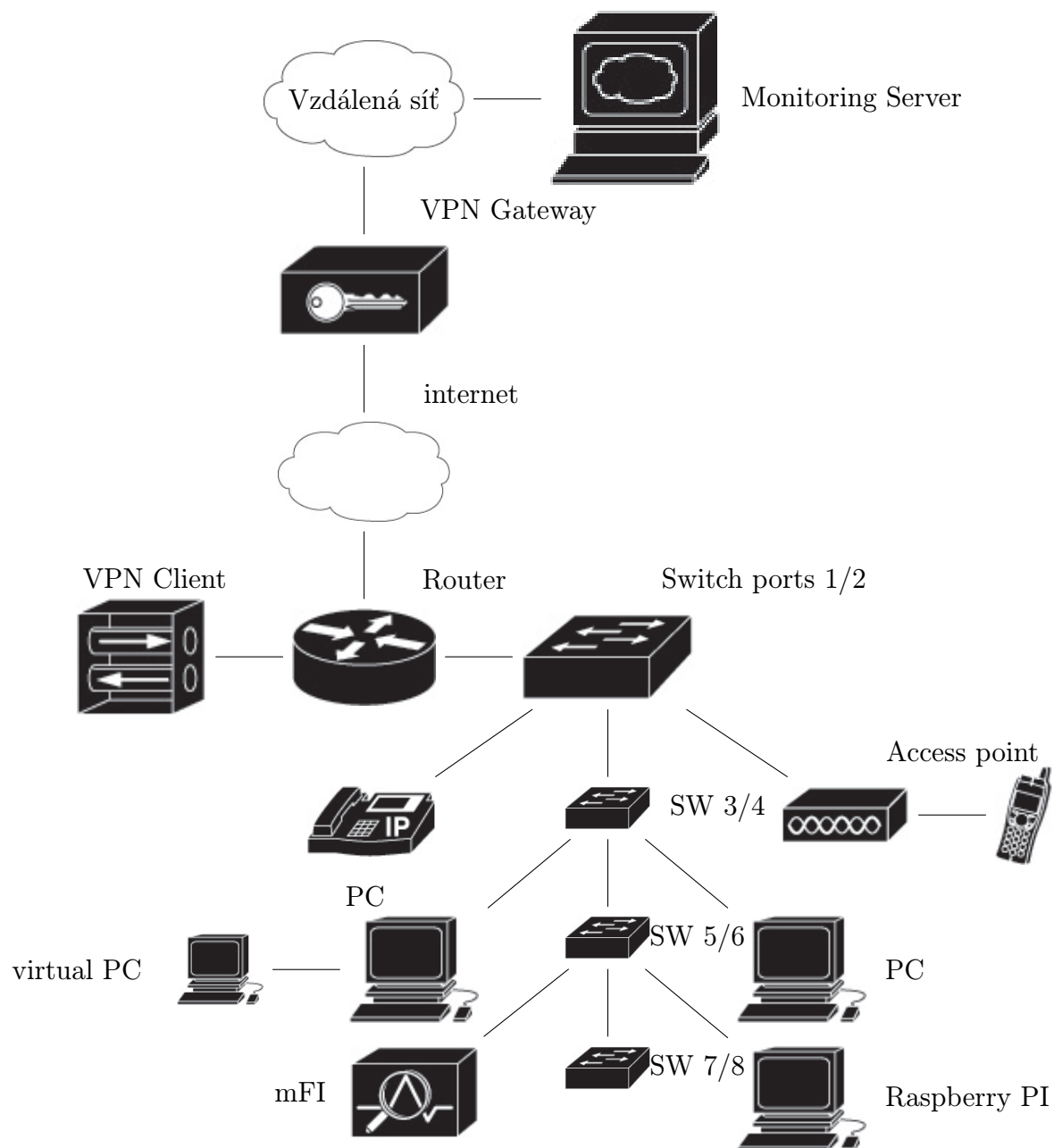
A	Monitorování několika nezávislých míst	71
B	Zapojení Síťových prvků	72
C	Měření Zabbix	73
D	Schémata – verze první	75
E	Desky plošných spojů – verze první	78
F	Náhled DPS – verze první	80
G	Schémata – verze druhá	81
H	Desky plošných spojů – verze druhá	85
I	Náhled DPS – verze druhá	89
J	Seznam součástek – verze první	90
K	Seznam součástek – verze druhá	91
L	Konfigurační soubory	93

## A MONITOROVÁNÍ NĚKOLIKA NEZÁVISLÝCH MÍST

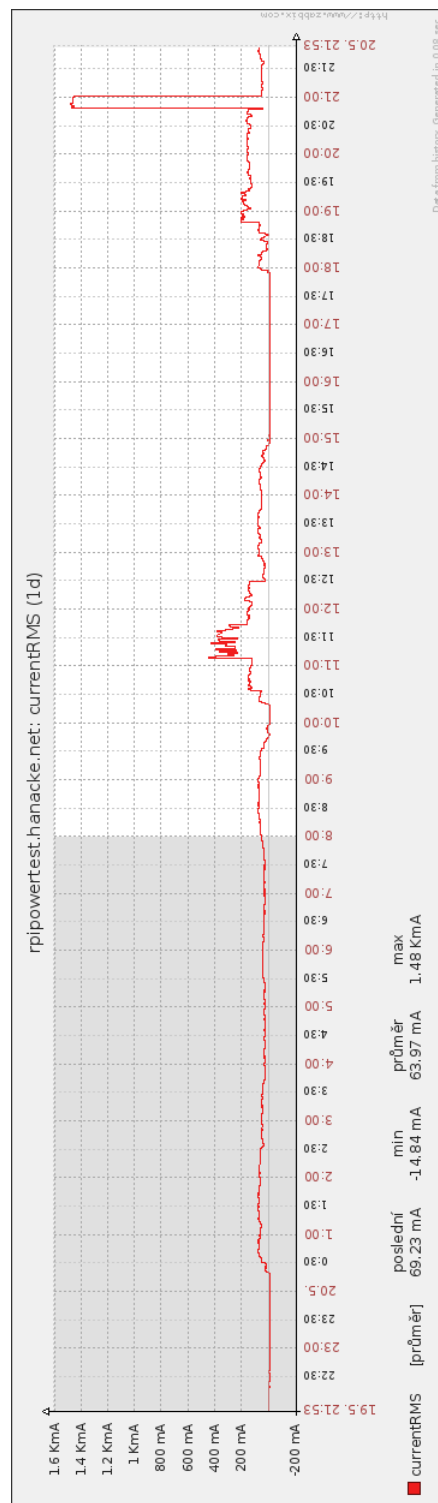
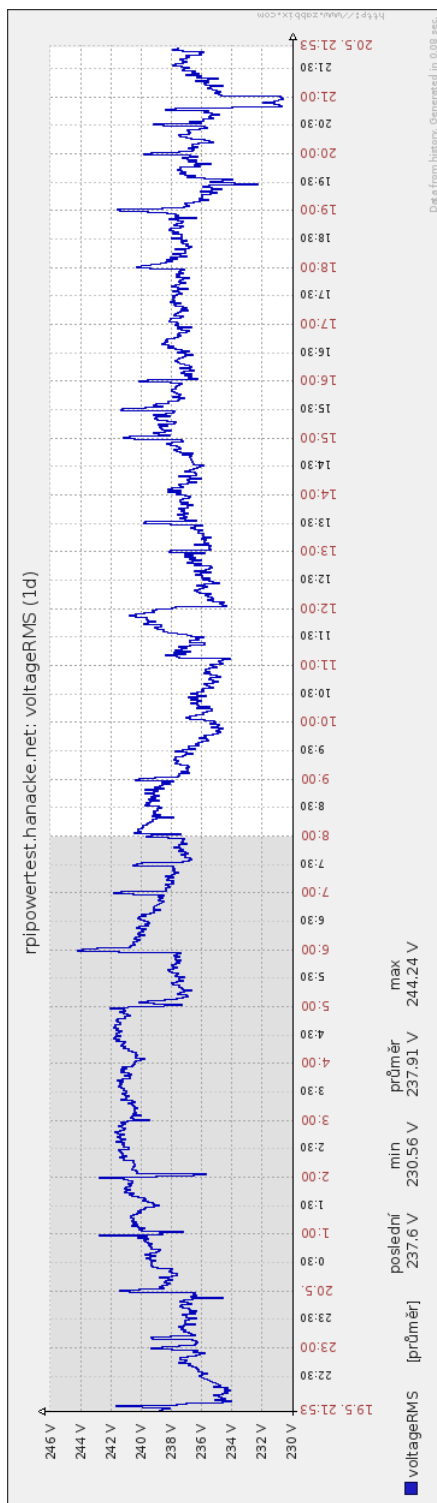




## B ZAPOJENÍ SÍŤOVÝCH PRVKŮ

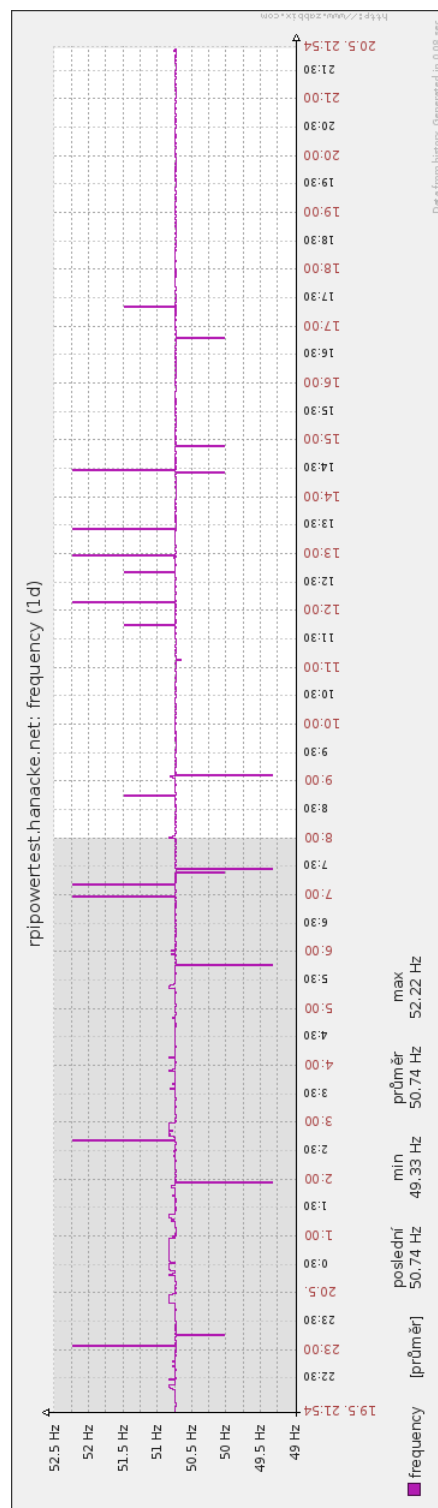
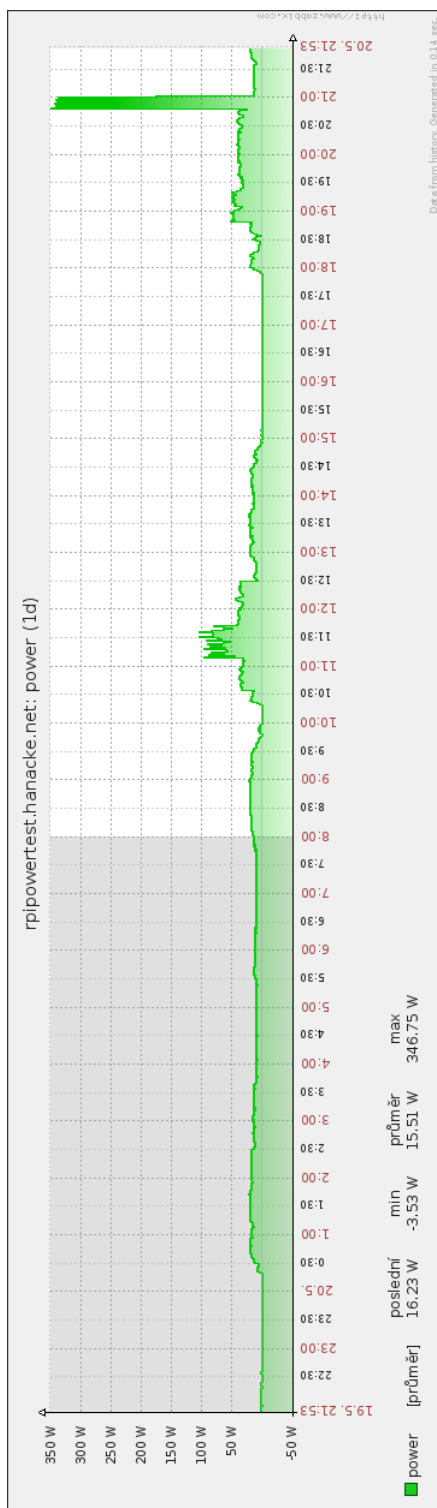


## C MĚŘENÍ ZABBIX



Obr. C.1: Průběh odebíraného napětí – měřeno systémem Zabbix

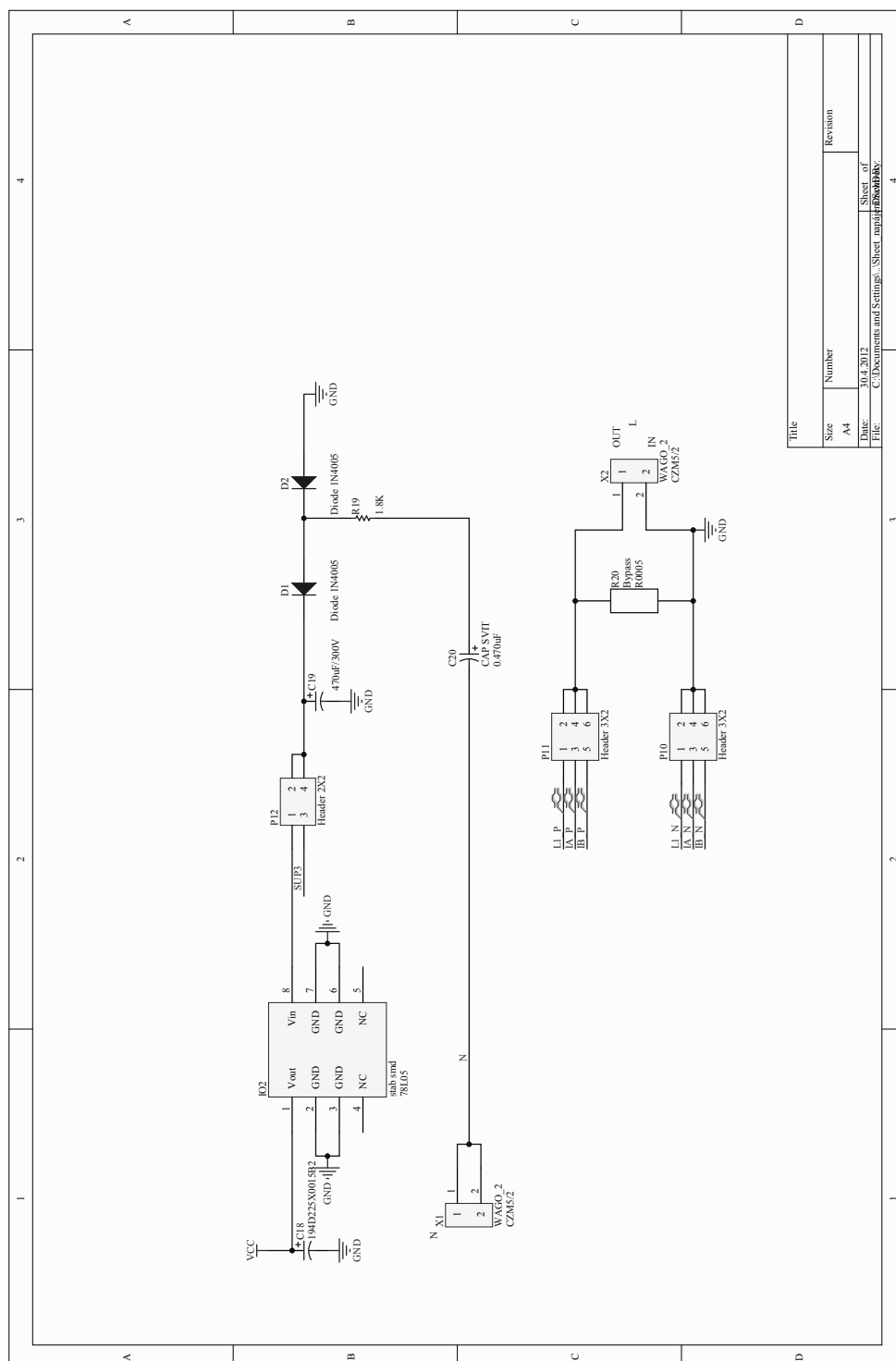
Obr. C.2: Průběh odebíraného proudu – měřeno systémem Zabbix



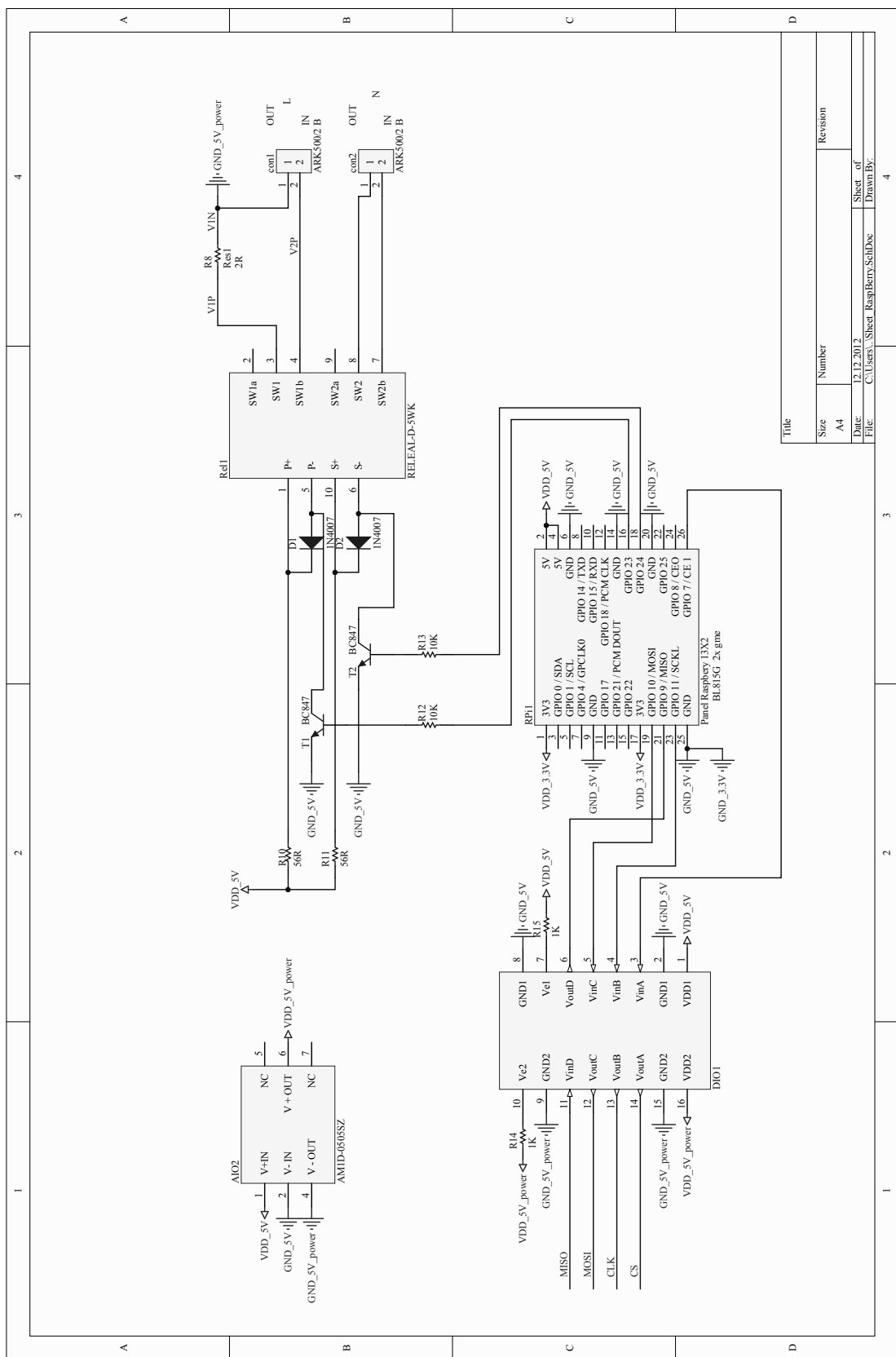
**Obr. C.3:** Průběh odebíraného výkonu – měřeno systémem Zabbix

**Obr. C.4:** Průběh síťové frekvence – měřeno systémem Zabbix

## D SCHÉMATA – VERZE PRVNÍ



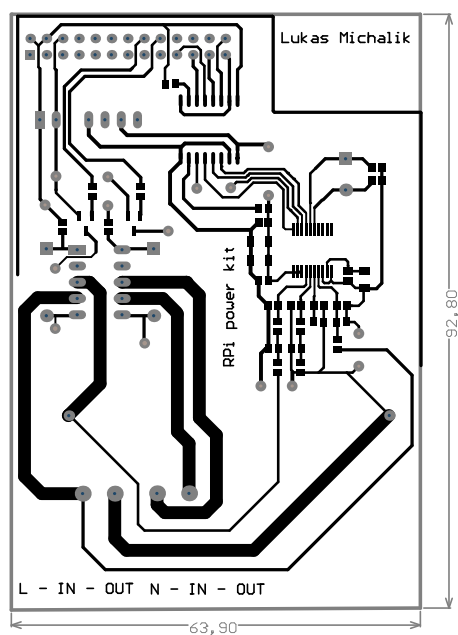
Obr. D.1: Schéma zapojení napájecího obvodu – (Tranformless)



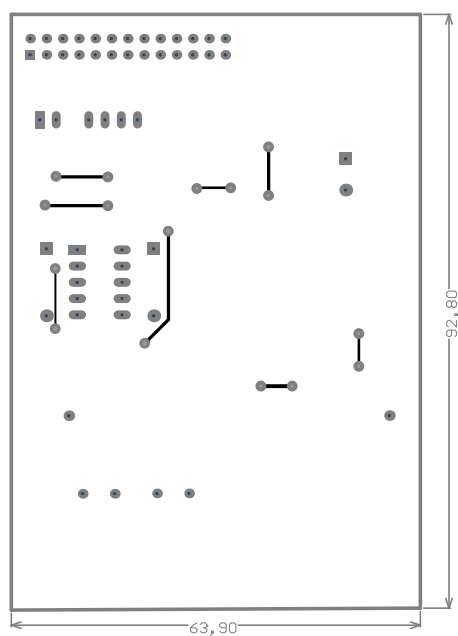
Obr. D.2: Schéma zapojení modulu pro Raspberry Pi – verze první



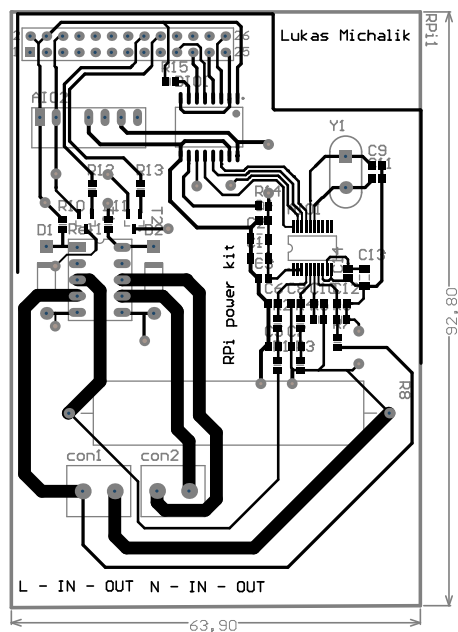
## E DESKY PLOŠNÝCH SPOJŮ – VERZE PRVNÍ



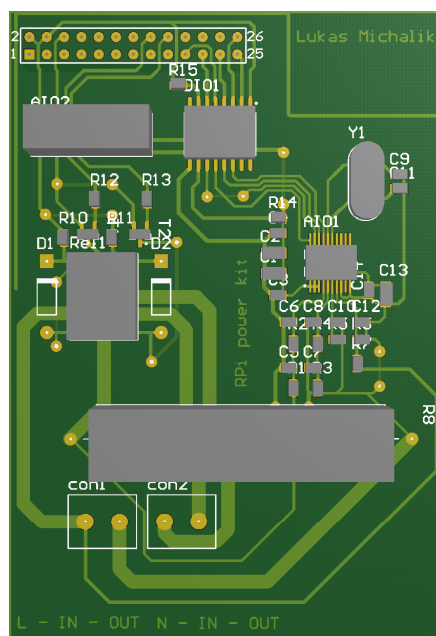
Obr. E.1: Deska plošných spojů – Top – verze první



Obr. E.2: Deska plošných spojů – Bottom – verze první



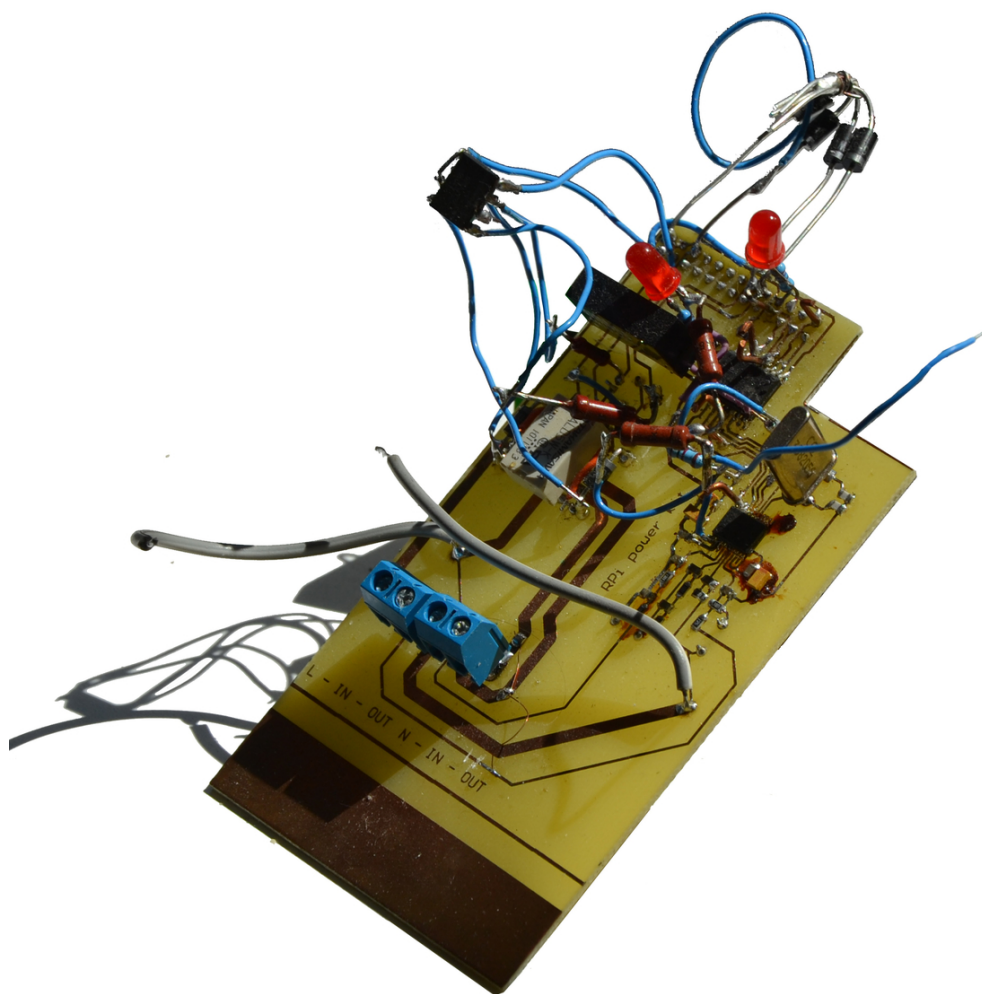
**Obr. E.3:** Deska plošných spojů – osazovací předloha – verze první



**Obr. E.4:** Deska plošných spojů – 3D náhled – verze první

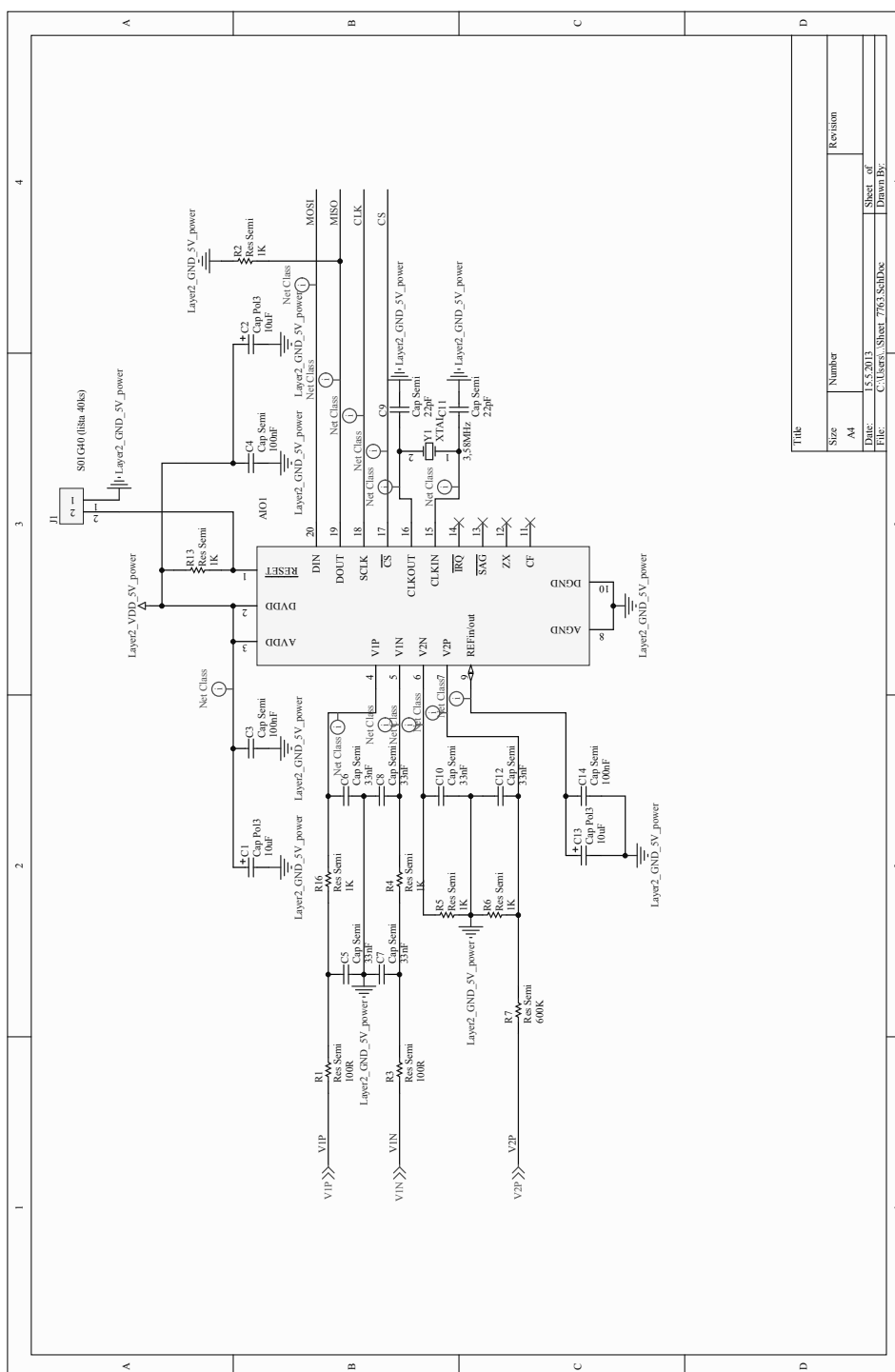


## F NÁHLED DPS – VERZE PRVNÍ



**Obr. F.1:** Náhled na první verzi měřicího modulu určeného pro Raspberry Pi – po dodatečných úpravách

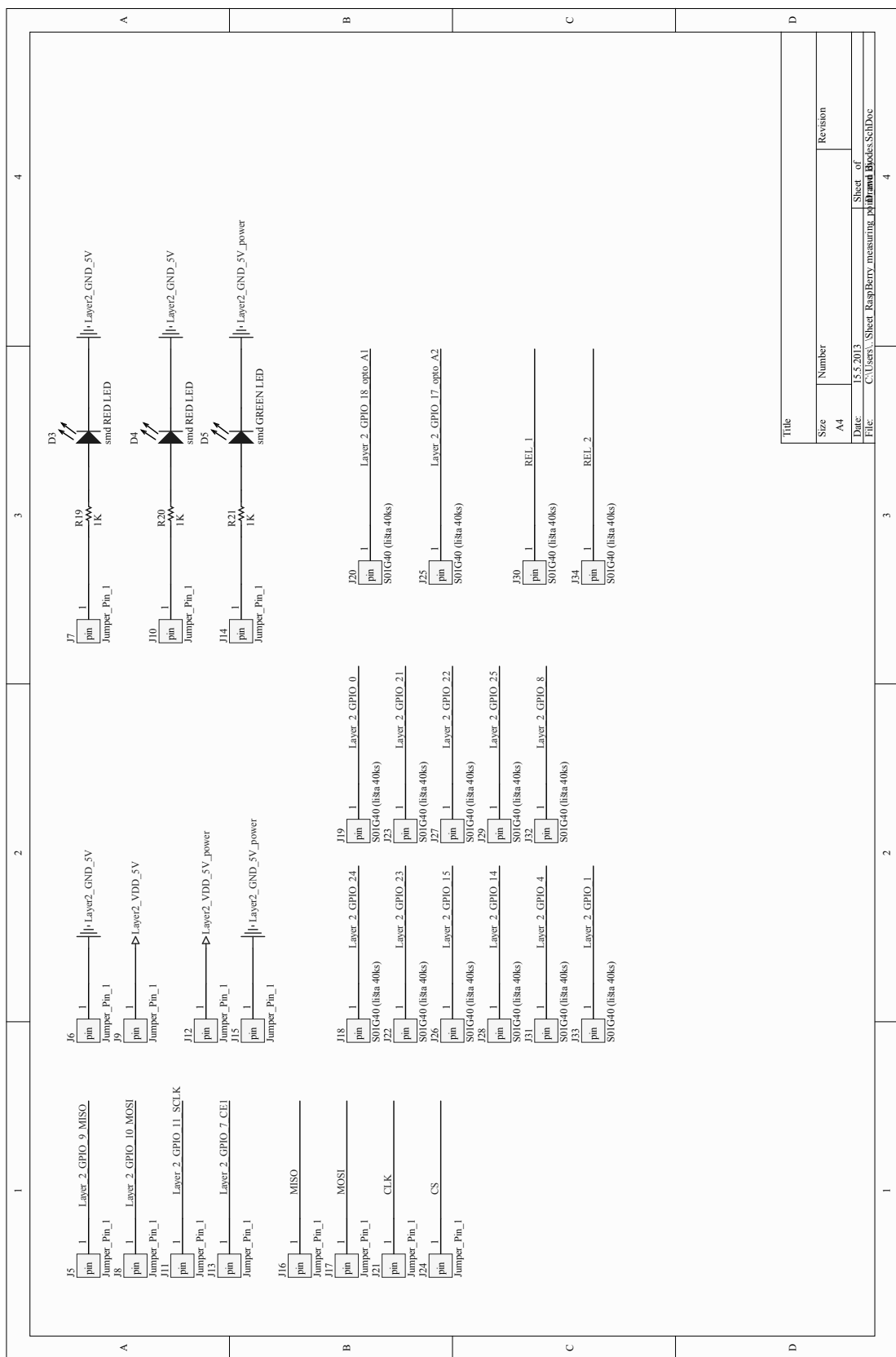
# G SCHÉMATA – VERZE DRUHÁ



Obr. G.1: Schéma zapojení obvodu ADE7763 – verze druhá

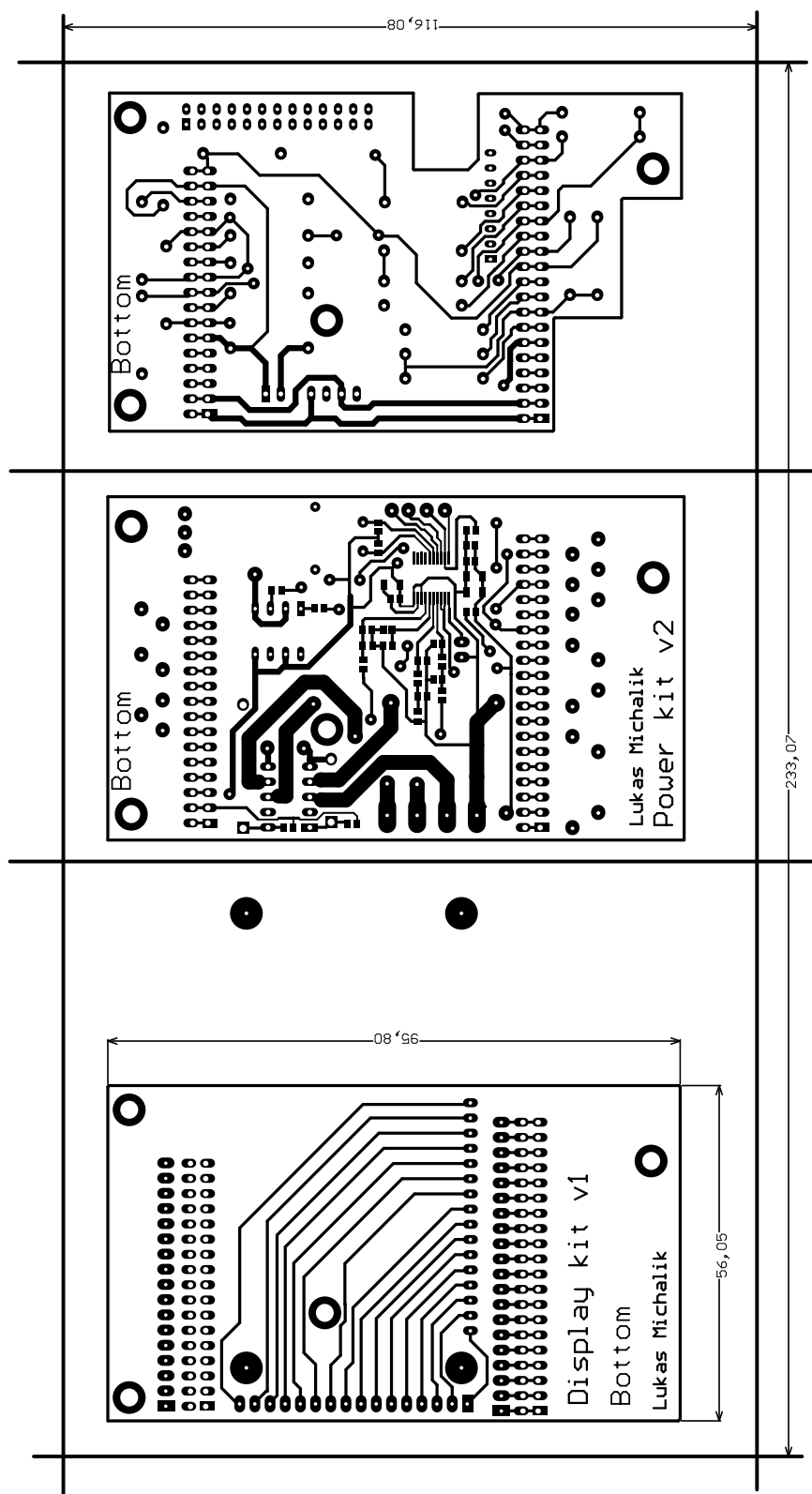






Obr. G.4: Schéma zapojení měřicích bodů a LED diod – verze druhá

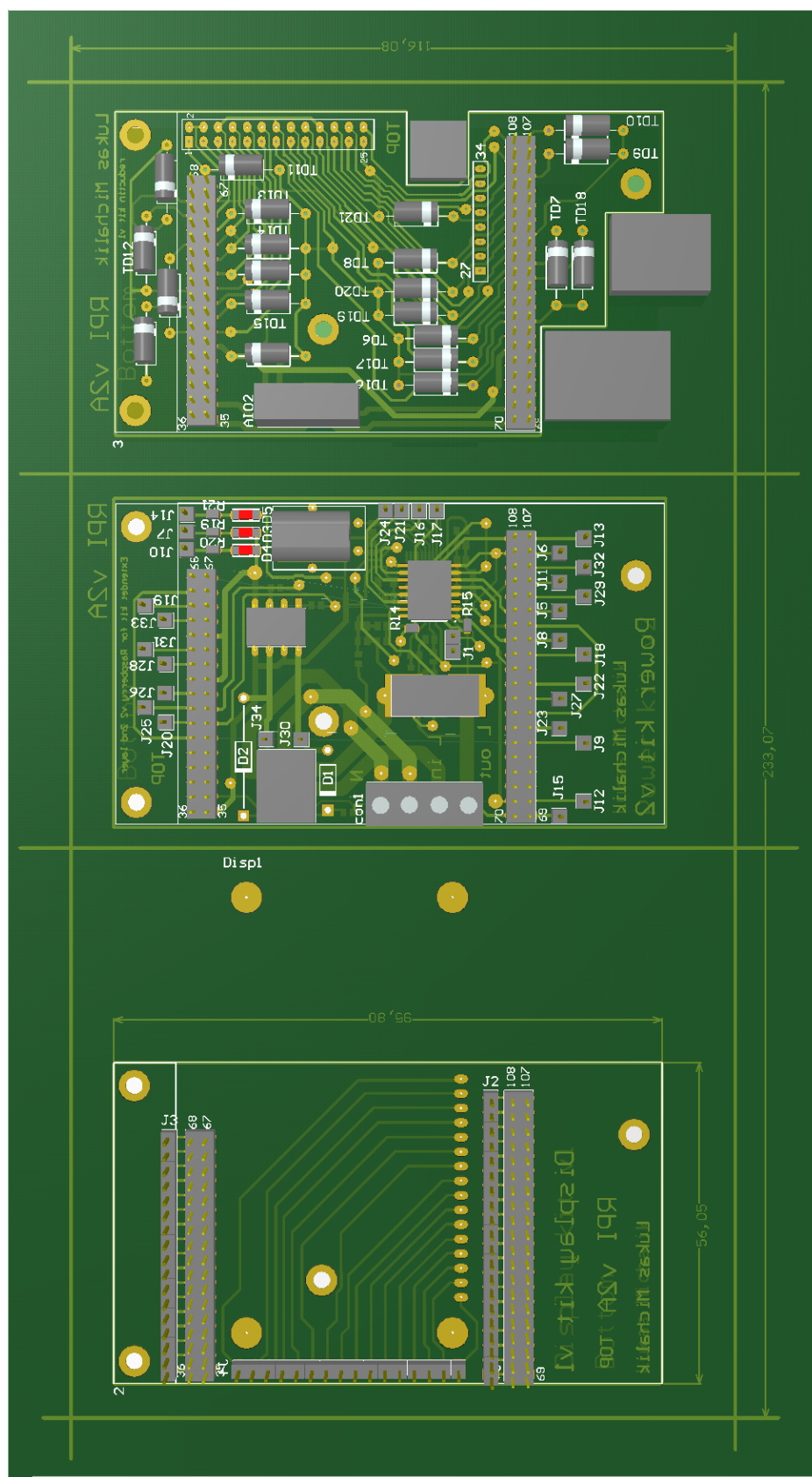




**Obr. H.2:** Deska plošných spojů – Bottom – verze druhá (zvětšení 0,75) – návrh obsahuje i rozšiřující modul pro LCD displej, ten však není v této práci využit

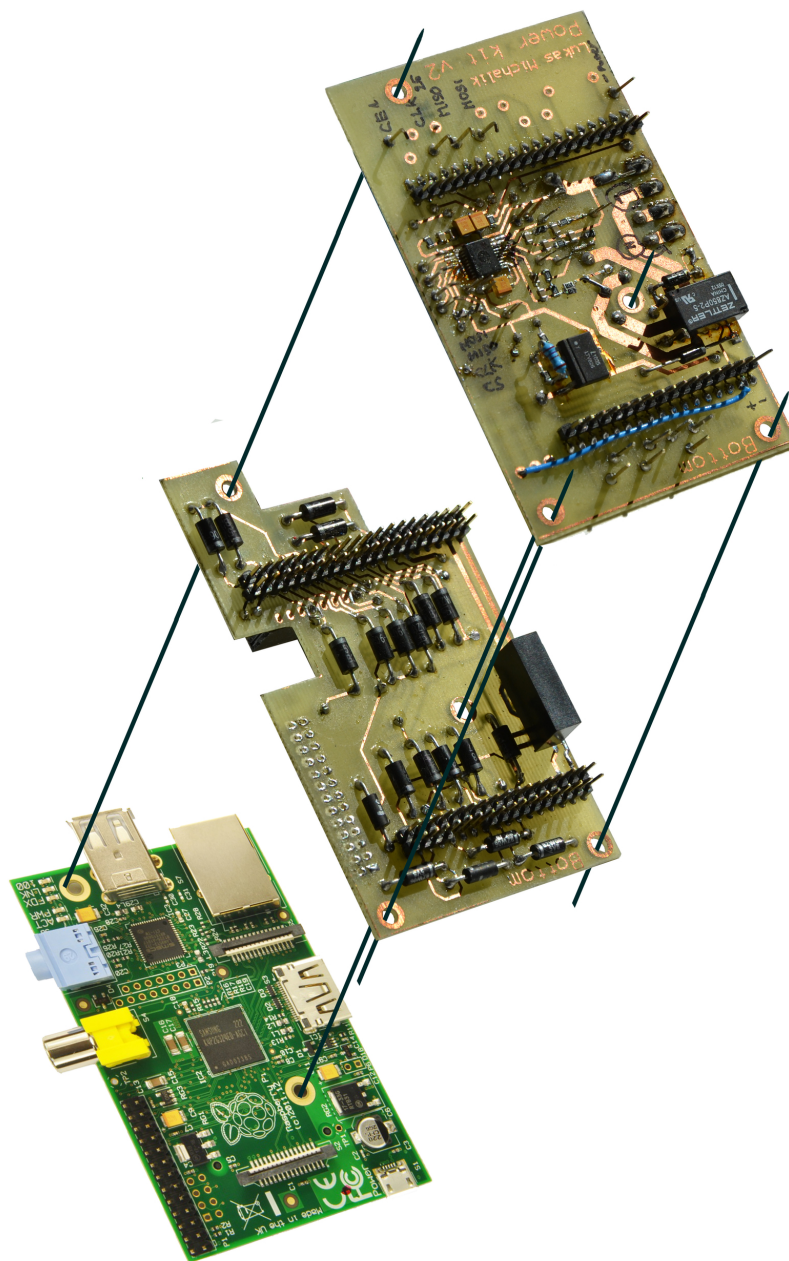






**Obr. H.4:** Deska plošných spojů – 3D náhled – verze druhá (zvětšení 0,75) – návrh obsahuje i rozšiřující modul pro LCD displej, ten však není v této práci využit

# I NÁHLED DPS – VERZE DRUHÁ



Obr. I.1: Náhled na druhou verzi měřicího modulu určeného pro Raspberry Pi

## J SEZNAM SOUČÁSTEK – VERZE PRVNÍ

**Tab. J.1:** Seznam použitých součástek – první verze měřicího modulu

Označení	Počet	Hodnota	Pouzdro
AIO1	1	ADE_7763	RS-20
AIO2	1	AM1D-0505SZ	Inline DIP 7
C1, C2, C13	3	10uF	CAPC3216L
C3, C4, C14	3	100nF	CAPC2012L
C5, C6, C7, C8, C10, C12	6	33nF	CAPC2012L
C9, C11	2	22pF	CAPC2012L
con1, con2	2	ARK500/2 B	ARK500
D1, D2	2	1N4007	DIO10.46-5.3x2.8
DIO1	1	ADuM_1401	RW-16 L
R1, R3	2	100R	RESC2012L
R2, R4, R5, R6, R14, R15	6	1K	RESC2012L
R7	1	1200K	RESC2012L
R8	1	2R	AXIAL-2.36
R10, R11	2	56R	RESC2012L
R12, R13	2	10K	RESC2012L
Rel1	1	RELEAL-D-5WK	Relay DIP 10
RPi1	1	BL815G 2x gme	HDR2X13
T1, T2	2	BC847	SOT23
Y1	1	3,58MHZ	HC49

## K SEZNAM SOUČÁSTEK – VERZE DRUHÁ

**Tab. K.1:** Seznam použitých součástek – druhá verze měřicího modulu

Označení	Počet	Hodnota	Pouzdro
OPT1	DIL8	1	LTV826
AIO1	RS-20	1	ADE 7763
AIO2	Inline DIP 7	1	AM1D-0505SZ
C1, C2, C13	CAPC3216L	3	10uF
C3, C4, C14	CAPC2012L	3	100nF
C5, C6, C7, C8, C10, C12	CAPC2012L	6	33nF
C9, C11	CAPC2012L	2	22pF
con1	ARK500/4	1	2x ARK500/2 B
D3, D4, D5	SMD 1206 (3216) RED	3	LED5R
DIO1	RW-16.L	1	ADuM 1401
Displ	Two Line 16 symbols	1	DATA IMAGE
J1, J2, J3, J4, J5, J6, J7, J8, J9, J10, J11, J12, J13, J14, J15, J16, J17, J18, J19, J20, J21, J22, J23, J24, J25, J26, J27, J28, J29, J30, J31, J32, J33, J34	Jumper	34	S01G40 (lišta 40ks)
R1, R3	RESC2012L	2	100R
R2, R4, R5, R6, R13, R14, R15, R16, R19, R20, R21	RESC2012L	11	1K
R7	RESC2012L	1	600K
R9	SMD 1575	1	R0005
R10, R11	RESC2012L	2	odstranit 0R
R17, R18	RESC2012L	2	120R
Rel1	Relay DIP 10	1	RELEAL-D-5WK
Pokračování na další straně			

Označení	Počet	Hodnota	Pouzdro
TD1, TD2, TD3, TD4, TD5, TD6, TD7, TD8, TD9, TD10, TD11, TD12, TD13, TD14, TD15, TD16, TD17, TD18, TD19, TD20, TD21	Transil 13mm	21	BZW06-5V8
Y1	HC49/U	1	3,58MHz
D1, D2	DIO	4	1N4007

# L KONFIGURAČNÍ SOUBORY

## Výpis: L.1: Konfigurační soubor snmpd

```
#####
#
# snmpd.conf
#
# - created by hand
# - man page http://net-snmp.sourceforge.net/docs/man/snmptrapd.conf.htm
#####
# SECTION: Access Control Setup
#
# This section defines who is allowed to talk to your running
# snmp agent.

###Configuring SNMP Version 2c Community

#directive      community      [source [OID]]
rocommunity     read_only_user  192.168.1.10 .1.3.6
rocommunity     read_only_user  192.168.1.15 .1.3.6
rocommunity     read_only_user  192.168.1.200 .1.3.6
rwcommunity     read_write_user 127.0.0.0

#requirement:
#snmpwalk -v2c -c read_only_user 192.168.1.10 system

###Configuring SNMP Version 3

#directive user [rwuser|rouser]  user name  Security Levels in SNMPv3
#           [noAuthnoPriv|authNoPriv|authPriv] [OID]
rwuser      userV3R0      authPriv
                                           .1

#requirement:
#snmpwalk -v3 192.168.1.10 system
#
# ~/.snmp/snmp.conf
#defVersion 3
#defSecurityLevel authPriv
#defSecurityName userV3R0
#defPassphrase 12345678

#####
# SECTION: Extend

#Extend nazev  program  1. parametr      2.parametr
#extend  cislo  /bin/sh  /home/lukas/skripts/test_snmp.sh  5
#extend datum /usr/bin/date
extend power /bin/sh /home/lukas/snmp_vitruual_spi/spi_power
extend avg_power /bin/sh /home/lukas/snmp_vitruual_spi/spi_avg_power
extend rele /bin/sh /home/lukas/snmp_vitruual_spi/proc_gpio
```

```

#requirement:
#snmpwalk -v3 192.168.1.10 NET-SNMP-EXTEND-MIB::nsExtendObjects

#####
# SECTION: System Information Setup
#
#   This section defines some of the information reported in
#   the "system" mib group in the mibII tree.

# syslocation: The [typically physical] location of the system.
#   Note that setting this value here means that when trying to
#   perform an snmp SET operation to the sysLocation.0 variable will make
#   the agent return the "notWritable" error code. IE, including
#   this token in the snmpd.conf file will disable write access to
#   the variable.
#   arguments:  location_string
syslocation Datacenter, Row 3, Rack 2

# syscontact: The contact information for the administrator
#   Note that setting this value here means that when trying to
#   perform an snmp SET operation to the sysContact.0 variable will make
#   the agent return the "notWritable" error code. IE, including
#   this token in the snmpd.conf file will disable write access to
#   the variable.
#   arguments:  contact_string
syscontact UNIX Admin <admin@example.com>

# syssservices: The proper value for the sysServices object.
#   arguments:  syssservices_number

syssservices 76

#####
# SECTION: Monitor Various Aspects of the Running Host
#
#   The following check up on various aspects of a host.

# proc: Check for processes that should be running.
#   proc NAME [MAX=0] [MIN=0]
#...
#   NAME:  the name of the process to check for. It must match
#           exactly (ie, http will not find httpd processes).
#   MAX:   the maximum number allowed to be running. Defaults to 0.
#   MIN:   the minimum number to be running. Defaults to 0.
#...
#   The results are reported in the prTable section of the UCD-SNMP-MIB tree
#   Special Case: When the min and max numbers are both 0, it assumes
#   you want a max of infinity and a min of 1.

#proc  sshd 1 1
#proc  crond 0 0
#requirement:
#snmpwalk -v3 192.168.1.10 UCD-SNMP-MIB::prTable

```

```

# disk: Check for disk space usage of a partition.
#   The agent can check the amount of available disk space, and make
#   sure it is above a set limit...
#...
#   disk PATH [MIN=100000]
#...
#   PATH:   mount path to the disk in question.
#   MIN:    Disks with space below this value will have the Mib's errorFlag set.
#           Can be a raw byte value or a percentage followed by the %
#           symbol. Default value = 100000.
#...
#   The results are reported in the dskTable section of the UCD-SNMP-MIB tree

#disk / 10000000
#disk /tmp 40000
disk /home 10000000
#requirement:
#snmpwalk -v3 192.168.1.10 UCD-SNMP-MIB::dskTable

#####
# SECTION: Trap Destinations
#
#   Here we define who the agent will send traps to.

# informsink: A~SNMPv2c inform (acknowledged trap) receiver
#   arguments: host [community] [portnum]
#               localhost [read_only_user] [6000]

# trapcommunity: Default trap sink community to use
#   arguments: community-string

trapcommunity read_only_user
# authtrapenable: Should we send traps when authentication failures occur
#   arguments: 1 | 2   (1 = yes, 2 = no)

authtrapenable 1

```

**Výpis: L.2:** Výpis zdrojového textu webové stránky zobrazující údaje o měření na Raspberry Pi

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs" dir="ltr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>RaspBerryPi power control</title>
<meta name="RaspBerryPi" content="Testpage"/>
<meta http-equiv="refresh" content="2" >
<link rel="shortcut icon" href="favicon.ico" />
</head>

<html><body><h1>RaspBerry Pi power control</h1>
<p>Testovací stránka pro měření spotřeby pomocí Raspberry Pi a přídavného
výkonového modulu.</p>
<p>Toto je pouze monitorovací zařízení. Neuchovává dlouhodobá data. Proto
nehledejte žádné grafy. Pro sběr

```



dat je možné využít protokol SNMP a monitorovací server. Například Zabbix (<http://www.zabbix.com/>)</p>

```
<table>
<tr>
<td> <h2>Napětí:</h2> <h1> <?php include("/mnt/powercontrol/Vrms");?> V </h1> </td>
<td> &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; </td>
<td> <h2>Proud:</h2> <h1> <?php include("/mnt/powercontrol/Irms");?> A </h1> </td>
<td> &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; </td>
<td> <h2>Spotřeba:</h2> <h1> <?php include("/mnt/powercontrol/Vykon");?> W </h1>
</td>
</tr>

<tr>
<td> <h2>Rele:</h2> <h1> <?php include("/mnt/powercontrol/Rele");?> </h1> 2 pro ON
3 proOFF</td>
<td> </td>
<td><h2>Frekvence:</h2> <h1> <?php include("/mnt/powercontrol/Frek");?> Hz
</h1></td>
<td> </td>
<td><h2></h2> <h1> </h1></td>
</tr>
</tr>
</table>

</body></html>
```

