

Diplomová práce



**Česká zemědělská univerzita  
v Praze**

**PEF**

Provozně ekonomická fakulta  
Katedra informačního inženýrství

# Monitorování výroby a regulace spotřeby elektrické energie pomocí technologie Raspberry Pi

**Bc. Jiří NAVRÁTIL**

Studijní obor: Informatika

Studijní program: Systémové inženýrství a informatika

Praha 2021

Vedoucí práce: Ing. Josef Pavlíček, Ph.D.

Děkuji RNDr. Petru Olšákovi za CSplain rozšíření plain T<sub>E</sub>Xu Donalda Knutha, za rozšiřující makra OPmac a za šablonu CTUstyle pro sazbu studentských závěrečných prací.

# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Jiří Navrátil

Systémové inženýrství a informatika

Informatika

Název práce

**Monitorování výroby a regulace spotřeby elektrické energie pomocí technologie Raspberry Pi**

Název anglicky

**Energy production monitoring and energy consumption regulation with Raspberry Pi**

---

### Cíle práce

Cílem práce je prakticky ověřit, zda je možné v rámci jednoho domu nasadit minipočítač Raspberry Pi 3 k monitorování výroby elektrické energie z fotovoltaické elektrárny, k monitorování stavu nabití akumulátorů a k návazné regulaci spotřeby elektrické energie.

Podcílím je pro monitorování a regulaci využít pouze lokálně dostupné prostředky. Tedy splnit cíl bez závislosti na cloudových službách. I při odpojení domu od Internetu musí být řešení nadále funkční.

### Metodika

Metodika diplomové práce je založena na nasazení technologie Raspberry Pi, analýze existujících zařízení pro chytrou domácnost a analýze u autora již nainstalované fotovoltaické elektrárny a jejího příslušenství.

- Analýza bude u Raspberry Pi a fotovoltaické elektrárny vycházet z dokumentace výrobců a u zařízení pro chytrou domácnost ze studia odborné literatury a již existujících hardwarových a softwarových řešení.
- Znalosti nabyté studiem budou zhodnoceny a na jejich základě bude vybrán a nasazen potřebný chybějící hardware a software. Pro propojení veškerého hardware a software do jednoho fungujícího celku realizujícího cíl a podcíl této práce bude navržena a implementována aplikace.
- V rámci vývoje bude použita inkrementální technika z agilní metodiky a pro testování bude využito přímé nasazení do reálného provozu.
- Řešení splňující cíl a podcíl musí být v provozu minimálně 2 měsíce před termínem pro odevzdání práce. Na základě výsledků z reálného provozu bude provedeno celkové zhodnocení a budou navrženy případné úpravy řešení.

**Doporučený rozsah práce**

60-70

**Klíčová slova**

Raspberry Pi (RPi), RaZberry, Z-Wave, fotovoltaická elektrárna (FVE)

---

**Doporučené zdroje informací**

PAETZ, Christian. Z-Wave Basics: Remote Control in Smart Homes. London: CreateSpace Independent Publishing Platform, 2013. ISBN 9781490537368.

---

**Předběžný termín obhajoby**

2020/21 LS – PEF

**Vedoucí práce**

Ing. Josef Pavlíček, Ph.D.

**Garantující pracoviště**

Katedra informačního inženýrství

Elektronicky schváleno dne 19. 11. 2020

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 19. 11. 2020

**Ing. Martin Pelikán, Ph.D.**

Děkan

V Praze dne 22. 11. 2020

## Poděkování / Prohlášení

Dovoluji si tímto poděkovat Ing. Josefu Pavlíčkovi, Ph.D. za vedení a cenné rady. Dále bych chtěl poděkovat mým dětem Dušanovi, Barborce, Kačence, Elišce a Pepíčkovi za podporu a pochopení, které mi věnovaly během tvorby této práce i celého studia.

Prohlašuji, že svou diplomovou práci „Monitorování výroby a regulace spotřeby elektrické energie pomocí technologie Raspberry Pi“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce.

V Praze dne 30. března 2021

.....

## Abstrakt / Abstract

V diplomové práci se autor věnuje optimalizaci spotřeby elektrické energie v rodinném domě za pomoci minipočítače Raspberry Pi. Dům má na střeše fotovoltaickou elektrárnu a ve sklepě systém elektrické akumulace. Přebytky vyrobené elektřiny jsou přednostně ukládány do akumulátorů a následně je upřednostněna vlastní spotřeba před odevzdáním do veřejné rozvodné sítě. V topné sezóně je spotřeba elektrické energie optimalizována ovládáním elektrického kotle ústředního topení.

Řízení takto pojaté chytré domácnosti je realizováno konzolovou aplikací, která na základě parametrů nastavených v konfiguraci a algoritmů implementovaných v kódu, řídí chod zapojených zařízení a o aktuálním stavu informuje v textové formě na terminálu a současně na Raspberry Pi pomocí LCD displeje.

Chod domácnosti je autonomní a nepotřebuje ke svému chodu připojení k Internetu. V případě jeho dostupnosti je možné skrze vzdálený zabezpečený ssh přístup sledovat aktuální stav a do chodu manuálně zasáhnout.

**Klíčová slova:** Raspberry Pi (RPi), RaZberry, fotovoltaická elektrárna (FVE), Z-Wave, Python, PostgreSQL, InfluxDB

This diploma thesis focuses on the optimization of electricity consumption in a family house using a Raspberry Pi minicomputer. The house has a photovoltaic power plant on the roof and an electrical storage system in the basement. Excess electricity, i.e. above the energy level required by the day to day running of the house itself, is initially stored in accumulators and is then used to power equipment which only runs intermittently, e.g. basement dehumidifiers. Any power above and beyond these requirements is then transferred to the external grid. During the colder months when heating is used, electricity consumption is optimized by controlling the electric central heating boiler.

The control of this model of smart home is managed via a console application which functions within the parameters which have been set in its configuration. As well as this, the code's algorithms control the operations of all connected devices and inform the user of the current status of these devices with regards to ambient temperatures in rooms, energy consumption, energy being generated and power purchased from the external grid. This appears in text form on the terminal and on Raspberry Pi via an LCD display.

The operation of the household is autonomous and does not require an Internet connection in order for it to function. If available, it is possible to monitor the live status of the house via a remote secure ssh access and intervene manually when necessary.

**Keywords:** Raspberry Pi (RPi), RaZberry, photovoltaic (PV) system, Z-Wave, Python, PostgreSQL, InfluxDB

# Obsah /

<b>1 Úvod</b> .....	1	4.7 PostgreSQL	48
<b>2 Cíl práce a metodika</b> .....	2	4.8 Zálohování a archivace	51
2.1 Cíl práce	2	4.9 Dokumentace	53
2.2 Metodika	2	4.10 Provozní údaje FVE	54
<b>3 Teoretická východiska</b> .....	3	<b>5 Zhodnocení výsledků a dopo-</b>	
3.1 Fotovoltaická elektrárna	3	<b>ručení</b> .....	56
3.1.1 Pojmy	3	<b>Závěr</b> .....	62
3.1.2 Historie	4	<b>Literatura</b> .....	64
3.1.3 Parametry použité FVE	5	<b>Seznam použitých zkratk</b>	67
3.1.4 Fronius aplikace	7	<b>Přílohy</b> .....	68
3.1.5 Rozhraní střídače	8		
3.1.6 Zapojení FVE	10		
3.2 Počítačová síť	10		
3.3 Raspberry Pi	12		
3.4 Chytrá domácnost	15		
3.5 Z-Wave	17		
3.5.1 RaZberry	17		
3.6 Testování	18		
3.6.1 Persony	18		
3.6.2 Metoda	19		
3.7 Programovací jazyk	19		
3.8 Verzovací systém	19		
3.9 Dokumentace	20		
3.10 Databáze	21		
3.10.1 Časové řady	21		
3.10.2 Relační databáze	21		
3.11 Hesla	21		
<b>4 Praktická část</b> .....	22		
4.1 Nákres domu	22		
4.2 Instalace řídicí jednotky	23		
4.3 Instalace Z-Wave zařízení	24		
4.4 Prvotní automatizace	29		
4.5 RaZberry Home	32		
4.5.1 InfluxDB	32		
4.5.2 Git	34		
4.5.3 Verze 1.0	35		
4.5.4 Modul rh (main)	36		
4.5.5 Modul config	38		
4.5.6 Modul device	40		
4.5.7 Modul room	41		
4.5.8 Modul hub	41		
4.5.9 Třída Dehumidifiers	42		
4.5.10 Třída Radiatory	42		
4.5.11 Inicializace objektů	43		
4.5.12 Použití aplikace	44		
4.6 Instalace LCD displeje	44		

## Tabulky / Obrázky

<b>3.1.</b> Fronius JSON API.....9	<b>3.1.</b> Podpora FVE pro rok 2017.....4
<b>3.2.</b> IPv4 adresy v LAN ..... 12	<b>3.2.</b> Podpora FVE pro rok 2021.....5
<b>3.3.</b> Velikost Git repozitáře..... 19	<b>3.3.</b> Fotovoltaické panely .....6
<b>3.4.</b> Makefile pro T <sub>E</sub> X..... 20	<b>3.4.</b> Střídač, baterie a jistič .....7
<b>4.1.</b> Maximální výroba v roce 2020 . 52	<b>3.5.</b> Tok energie .....8
	<b>3.6.</b> ZyXEL PowerLine Ethernet .....9
	<b>3.7.</b> Diagram zapojení FVE ..... 10
	<b>3.8.</b> ASUS Lyra ..... 12
	<b>3.9.</b> GPIO rozhraní ..... 14
	<b>3.10.</b> Odvlhčovače ..... 16
	<b>3.11.</b> Z-Wave Plus ..... 17
	<b>3.12.</b> RaZberry na Raspberry Pi .... 18
	<b>4.1.</b> Nákres domu ..... 23
	<b>4.2.</b> Z-Wave termostat ..... 25
	<b>4.3.</b> Monočlánky nabité na 55%.... 25
	<b>4.4.</b> Monočlánky nabité na 5% ..... 26
	<b>4.5.</b> Z-Wave zásuvka ..... 27
	<b>4.6.</b> Z-Wave spínač topení ..... 27
	<b>4.7.</b> Z-Wave dashboard ..... 28
	<b>4.8.</b> Raspberry Pi a LCD displej ... 48
	<b>4.9.</b> Vyrobená elektřina celkem ..... 55
	<b>5.1.</b> Ostrovní režim ..... 59
	<b>5.2.</b> Hranice minimálního nabití.... 59
	<b>5.3.</b> Výroba 22. 9. 2020 ..... 60
	<b>5.4.</b> Výroba 21. 7. 2020 ..... 60
	<b>5.5.</b> Hustota mraků ..... 61
	<b>A.1.</b> Parametry střídače 1/2 ..... 69
	<b>A.2.</b> Parametry střídače 2/2 ..... 70
	<b>A.3.</b> Smart Meter..... 70
	<b>A.4.</b> Parametry baterie..... 71
	<b>C.5.</b> Celkově vyrobená elektřina .... 73
	<b>C.6.</b> Vyrobená elektřina dle let ..... 73
	<b>C.7.</b> Demontované solární panely ... 74
	<b>C.8.</b> Certifikát 24 hodin slunce ..... 75
	<b>D.9.</b> Výroba 2017 dle využití..... 76
	<b>D.10.</b> Výroba 2018 dle využití..... 76
	<b>D.11.</b> Výroba 2019 dle využití..... 77
	<b>D.12.</b> Výroba 2020 dle využití..... 77
	<b>E.13.</b> Faktura 2020/07..... 78
	<b>H.14.</b> Solar.web iOS aplikace..... 82
	<b>K.15.</b> Tesla Model 3 ..... 86



# Kapitola 1

## Úvod

Ve stejné době, kdy jsem přemýšlel nad tématem má diplomové práce, jsem také zvažoval, jak co nejlépe využít moji fotovoltaickou elektrárnu. Oba myšlenkové proudy se v hlavě setkaly a já se rozhodl spojit moji závěrečnou práci s řešením toho, co jsem potřeboval a chtěl vyřešit. Postupně tak začala vznikat tato práce, ve které jsem využíval dříve i nově získané znalosti a s jejich postupným nasazováním jsem je ověřoval v praxi.

V tomto okamžiku, kdy moji práce čtete, je moje řešení plně funkční. Cíl i podcíl práce jsou splněné. Maličký deskový počítač Raspberry Pi monitoruje a řídí spotřebu elektrické energie. V textu dále je popsáno, jak je moje pojetí chytré domácnosti řešeno, jaká zařízení používám a jakým způsobem vše funguje dohromady jako jeden chytrý celek.

Cesta, po které jsem se vydal, nebyla vždy jednoduchá. Musel jsem nejprve z existenčních důvodů a později ze zdravotních důvodů kombinované studium dvakrát přerušit. Přesto byly přestávky na práci minimální a řešení je již řadu měsíců v provozu. Dům se nachází na Suchdole a od mé fakulty je vzdálen vzdušnou čarou 1,11 km. Rád řešení ještě před obhajobou či následně po obhajobě předvedu osobně na místě. Dále mohu přímo při obhajobě realizovat z notebooku vzdálené připojení, ukázat aktuální výstupy ze živého provozu a případně do něho ručně zasáhnout.

Moje diplomová práce, kterou držíte v ruce, je odevzdaná. Práce na řešení jako taková však nekončí. Průběžně implementuji různá zlepšení. Nadále se učím novým věcem a některé z nich přenáším do kódu, který jsem v rámci této práce vytvořil a dále ho vylepšuji. Práce je pro mne vzorovým kódem pro použití objektového návrhu v jazyce Python, ukládání dat do relační databáze PostgreSQL a zapisování časových řad do databáze InfluxDB. Očekávám připomínky z oponentury a od komise. Rád se z nich poučím a nasadím do praxe.

Do příloh, tedy mimo počítaný rozsah práce, jsem zaznamenal postřehy a náměty, které by stály za další probádání či jako inspirace nebo jako poučení z chyb, kterých jsem se já během tvorby práce dopustil.

Děkuji, že jste můj úvod dočetli až sem. Budu rád, když pro vás bude moje práce užitečná. Těším se na její představení u obhajoby. Rád si o ní pohovořím i mimo ni.

*Jiří Navrátil*

# Kapitola 2

## Cíl práce a metodika

Tato kapitola přebírá v nezměněné formě cíl, podcíl a metodiku ze schváleného zadání diplomové práce.

### 2.1 Cíl práce

Cílem práce je prakticky ověřit, zda je možné v rámci jednoho domu nasadit minipočítač Raspberry Pi 3 k monitorování výroby elektrické energie z fotovoltaické elektrárny, k monitorování stavu nabití akumulátorů a k návazné regulaci spotřeby elektrické energie.

Podcílem je pro monitorování a regulaci využít pouze lokálně dostupné prostředky. Tedy splnit cíl bez závislosti na cloudových službách. I při odpojení domu od Internetu musí být řešení nadále funkční.

### 2.2 Metodika

Metodika diplomové práce je založena na nasazení technologie Raspberry Pi, analýze existujících zařízení pro chytrou domácnost a analýze u autora již nainstalované fotovoltaické elektrárny a jejího příslušenství.

Analýza bude u Raspberry Pi a fotovoltaické elektrárny vycházet z dokumentace výrobců a u zařízení pro chytrou domácnost ze studia odborné literatury a již existujících hardwarových a softwarových řešení.

Znalosti nabyté studiem budou zhodnoceny a na jejich základě bude vybrán a nasazen potřebný chybějící hardware a software. Pro propojení veškerého hardware a software do jednoho fungujícího celku realizujícího cíl a podcíl této práce bude navržena a implementována aplikace.

V rámci vývoje bude použita inkrementální technika z agilní metodiky a pro testování bude využito přímé nasazení do reálného provozu.

Řešení splňující cíl a podcíl musí být v provozu minimálně 2 měsíce před termínem pro odevzdání práce. Na základě výsledků z reálného provozu bude provedeno celkové zhodnocení a budou navrženy případné úpravy řešení.

# Kapitola 3

## Teoretická východiska

Základy této práce byly položeny dlouho před tím, než jsem poprvé otevřel textový editor Vim a napsal první řádek této práce.

Moje nadšení do zelené energie a elektromobility dostalo v roce 2016 silný impuls. V České republice začaly platit nové lepší podmínky pro instalaci solárních elektráren (viz podkapitola 3.1.2). V zahraničí Elon Musk ukázal prototyp elektromobilu Tesla Model 3. Tesla zakoupila SolarCity a začala mluvit o solárních taškách a představila novou verzi baterií Powerwall 2.0 vylepšených pro spolupráci s plánovanými solárními taškami.

Začal jsem zvažovat, zda si pořídit fotovoltaickou elektrárnu. Jako student České zemědělské univerzity jsem si uvědomoval, že umístění fotovoltaické elektrárny na střechu domu nemá negativní dopad na krajinu a na množství úrodné půdy. Zjišťoval jsem tedy, zda bude možné na můj dům fotovoltaickou elektrárnu umístit, zda se mi podaří zajistit financování formou úvěru a zda mám nárok na dotaci. Všechny tři odpovědi byly kladné, a tak jsem si fotovoltaickou elektrárnu pořídil.

V textu dále uvádím informace potřebné pro samotnou práci. Problematika podmínek umísťování a financování fotovoltaických elektráren v této práci není dále rozpracována. Toto téma by si zasloužilo samostatnou analýzu.

Dalším východiskem pro moji práci byla již existující infrastruktura v domě. Jako živnostník preferující fyzický hardware před využíváním cloudu jsem měl pro přípravu a realizaci této práce již vybudovanou síťovou i počítačovou infrastrukturu, kterou bylo možné ihned nasadit. V textu dále popisuji co, proč a jak jsem v práci použil a případně doplnil.

V této kapitole kromě teoretických východisek uvádím i další východiska. Název kapitoly jsem ponechal dle osnovy [24] a přidal jsem praktická východiska, která byla na počátku práce. Praktická část navazuje na teoretické poznatky z této kapitoly a na existující (praktická) východiska.

### 3.1 Fotovoltaická elektrárna

V této podkapitole definuji základní pojmy z oblasti fotovoltaiky, popisuji legislativní změny v roce 2016 a specifikuji parametry u mne nainstalované fotovoltaické elektrárny.

#### 3.1.1 Pojmy

Metodická pomůcka Ministerstva pro místní rozvoj [10] k pojmu *fotovoltaika* uvádí, že pochází ze dvou slov, řeckého  $\varphi\omega\varsigma$  „phos“ = světlo a ze jména italského fyzika Alessandra Volty.

Pro *fotovoltaický jev* zmiňuje, že ho objevil Alexandr Edmond Becquerel, který si v roce 1839 při pokusech se dvěma kovovými elektrodami umístěnými v elektrovodivém roztoku všiml, že při osvětlení zařízení vzrostlo na elektrodách napětí.

### 3.1.2 Historie

K historii využívání solární energie dříve uvedený materiál [10] doplňuje, že rozvoj fotovoltaiky nastartovali *kosmický výzkum* (solární články jako výhodný zdroj energie pro vesmírné družice), *celosvětová ropná krize* v roce 1973 a potřeba *čisté energie z obnovitelných zdrojů*.

Historie podporování a nepodporování energie z obnovitelných zdrojů v České republice je téma na samostatnou práci. Pro představu o vývoji výkupních cen elektřiny s využitím slunečního záření pro provozovny uvedené do provozu před rokem 2014 přikládám dvě cenová rozhodnutí Energetického regulačního úřadu. ERÚ tak činí na základě Zákona č. 526/1990 Sb., o cenách, Zákona č. 265/1991 Sb., o působnosti orgánů České republiky v oblasti cen, Zákona č. 458/2000 Sb., o podmínkách podnikání a o výkonu státní správy v energetických odvětvích a o změně některých zákonů (energetický zákon) a Zákona č. 165/2012 Sb., o podporovaných zdrojích energie a o změně některých zákonů. V obrázku 3.1 z Cenového rozhodnutí Energetického regulačního úřadu č. 9/2016 ze dne 14. prosince 2016 [15] a v obrázku 3.2 z Cenového rozhodnutí Energetického regulačního úřadu č. 7/2020, ze dne 30. září 2020 [16] je patrné, že na výkupní ceny elektřiny vyrobené využitím slunečního záření má vliv zejména rok uvedení do provozu a dále instalovaný výkon.

ř./sl.	Podporovaný druh energie	Datum uvedení výroby do provozu		Instalovaný výkon výroby [kW]		Jednotarifní pásmo provozování	
		od (včetně)	do (včetně)	od	do (včetně)	Výkupní ceny [Kč/MWh]	Zelené bonusy [Kč/MWh]
a	b	c	d	e	l	m	
501	Výroba elektřiny využitím slunečního záření	1.1.2006	31.12.2007	-	-	16 518	15 918
502		1.1.2008	31.12.2008	-	-	16 110	15 510
503		1.1.2009	31.12.2009	0	30	15 115	14 415
504		1.1.2009	31.12.2009	30	-	15 004	14 404
505		1.1.2010	31.12.2010	0	30	14 077	13 377
506		1.1.2010	31.12.2010	30	-	13 966	13 366
507		1.1.2011	31.12.2011	0	30	8 446	7 746
508		1.1.2011	31.12.2011	30	100	6 647	6 047
509		1.1.2011	31.12.2011	100	-	6 194	5 594
510		1.1.2012	31.12.2012	0	30	6 802	6 102
511		1.1.2013	30.6.2013	0	5	3 691	2 991
512		1.1.2013	30.6.2013	5	30	3 064	2 364
513		1.7.2013	31.12.2013	0	5	3 236	2 536
514		1.7.2013	31.12.2013	5	30	2 632	1 932

**Obrázek 3.1.** Cenové rozhodnutí Energetického regulačního úřadu č. 9/2016

ř./sl.	Podporovaný druh energie	Datum uvedení výroby do provozu		Instalovaný výkon výroby [kW]		Jednotarifní pásmo provozování	
		od (včetně)	do (včetně)	od	do (včetně)	Výkupní ceny [Kč/MWh]	Zelené bonusy [Kč/MWh]
		b	c	d	e	l	m
500	Výroba elektřiny využitím slunečního záření	-	31.12.2005	-	-	8 520	7 625
501		1.1.2006	31.12.2007	-	-	17 880	16 985
502		1.1.2008	31.12.2008	-	-	17 438	16 543
503		1.1.2009	31.12.2009	0	30	16 361	15 466
504		1.1.2009	31.12.2009	30	-	16 240	15 346
505		1.1.2010	31.12.2010	0	30	15 238	14 343
506		1.1.2010	31.12.2010	30	-	15 117	14 223
507		1.1.2011	31.12.2011	0	30	9 142	8 247
508		1.1.2011	31.12.2011	30	100	7 195	6 300
509		1.1.2011	31.12.2011	100	-	6 704	5 810
510		1.1.2012	31.12.2012	0	30	7 363	6 469
511		1.1.2013	30.06.2013	0	5	3 995	3 101
512		1.1.2013	30.06.2013	5	30	3 317	2 422
513		1.7.2013	31.12.2013	0	5	3 503	2 608
514		1.7.2013	31.12.2013	5	30	2 850	1 955

**Obrázek 3.2.** Cenové rozhodnutí Energetického regulačního úřadu č. 7/2020

Údaje na obrázku 3.1 a na obrázku 3.2 se týkají registrovaných podnikatelů, kteří své výroby elektřiny s využitím slunečního záření připojili k rozvodné síti před rokem 2014 a museli splnit přísné podmínky uvedené mimo jiné v již citované Metodické pomůcce *Ministerstva pro místní rozvoj s pokyny k umístování, povolování a užívání fotovoltaických staveb a zařízení* [10].

Pořízení a připojení mojí fotovoltaické elektrárny předcházely legislativní změny, které zjednodušily od roku 2016 vybudování domácích fotovoltaických elektráren do výkonu 10 kW a jejich připojování do rozvodné sítě. Konkrétně díky Zákonu č. 131/2015 Sb. [11] od 1. ledna 2016 odpadly pro zájemce nepodnikatele při pořizování fotovoltaické elektrárny tyto povinnosti:

- licence ERÚ
- výkazy vůči OTE a ERÚ
- živnostenské oprávnění
- daňová evidence
- stavební povolení
- kolaudační rozhodnutí

Tímto bych uzavřel východiska předcházející pořízení mojí fotovoltaické elektrárny. Pro zájemce o další informace o této problematice doporučuji hledat pojmy „26% solární daň“, „výkupní ceny elektřiny“, „ostrovní systémy bez zelených bonusů“ a „závazek České republiky vůči Evropské unii vyrábět z čistých zdrojů 13% energie“.

### ■ 3.1.3 Parametry použité FVE

Pro dodávku a instalaci fotovoltaické elektrárny jsem vybral PREměření. Bydlím v Praze na území spravovaném firmou PREdistribuce, jsem zákazníkem PRE, cenová nabídka PREměření byla podobná ostatním nabídkám a při mém požadavku na tehdy nejvýkonější panely mi PREměření vyhovělo a dodali mnou požadované panely při nezměněné ceně (informace o ceně viz příloha J). Jednalo se o panely AUO Solar SunForte PM096B00 o výkonu 330 Wp (detailní specifikace viz příloha B).

Na mém rodinném domě je rovná střecha. Chtěl jsem dosáhnout maximálního možného výkonu, a tak byla střecha osazena celá. Díky rovné střeše bylo možné nasměrování

i sklon panelů nastavit do optimální polohy vůči slunci. Jejich rozmístění je vidět na fotografii z dronu na obrázku 3.3. Celkem se na střechu vešlo 18 kusů solárních panelů a dohromady měly dle výrobce zvládnout maximální výkon 5,94 kWp. Dle E.ON Rádce [17] „fotovoltaické panely s celkovým výkonem 1 kWp vyrobí v našich podmínkách ročně zhruba 950 kWh elektrické energie“, odpovídá u mé instalace s výkonem 5,94 kWp výroba zhruba 5643 kWh elektrické energie<sup>1</sup>. Za provozu zvládly panely tento výkon překonat viz tabulka 4.1 v praktické části a také příloha I.



**Obrázek 3.3.** Fotovoltaické panely na střechě domu

K další důležité vlastnosti těchto panelů je možnost provozování u letiště. Bydlím nedaleko Mezinárodního letiště Václava Havla ve směru přistávací a vzletové dráhy 06/24 nasměrované severovýchod-jihozápad. Zvolené panely jsou vhodné díky svým vlastnostem pro provoz na letišti a v blízkosti letiště. Díky svým parametrům (viz příloha B) jsou provozovány například ve Spolkové republice Německo na Mezinárodním letišti Düsseldorf a já jsem pro provoz na Suchdole obdržel souhlas od Řízení letového provozu České republiky (viz příloha F).

Pro ukládání přebytků vyrobené elektřiny bylo nainstalováno šest bateriových dílců Fronius<sup>2</sup> s kapacitou nominální 1,5 kWh a užitečnou 1,2 kWh. V součtu tedy dávají celkovou nominální kapacitu 9 kWh a užitečnou kapacitu 7,2 kWh<sup>3</sup>.

K řízení provozu byl nainstalován hybridní střídač Fronius SYMO 5.0-3-S. Jeho úkolem je řídit toky elektrické energie. Před započítím této práce byl jediným řídicím prvkem právě tento střídač.

<sup>1</sup> 5,94 kWp \* 950 kWh = 5643 kWh

<sup>2</sup> Na popiskám je uveden jako výrobce SONY.

<sup>3</sup> To je maximální reálné dosažitelná a také opakovaně dosažená hodnota.

Instalovaný systém je schopný ostrovního režimu. Střídač ve spolupráci s jističem reaguje na přerušení dodávky elektřiny z rozvodné sítě přechodem do ostrovního režimu. Střídač potřebuje pro přechod do ostrovního režimu přibližně minutový restart<sup>1</sup>, během kterého není přerušena dodávka elektrického proudu nahrazena z vlastních zdrojů. Následně je v ostrovním režimu provoz domácnosti uspokojován ze solárních panelů a z baterie.



**Obrázek 3.4.** Jistič, bojler, elektrokotel, baterie a střídač

Na obrázku 3.4 ze sklepa je zleva doprava vidět jistič, bojler, elektrický kotel, baterie, Z-Wave spínač (viz také obrázek 4.6 v podkapitole 4.2) a střídač.

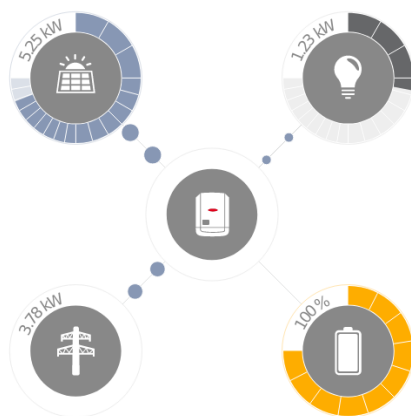
### ■ 3.1.4 Fronius aplikace

Výrobce střídače a baterií společnost Fronius International GmbH s centrálou ve Froniusplatz 1, 4600 Wels, Rakousko provozuje na adrese <https://www.solarweb.com/> webovou aplikaci, která umožňuje registraci produktů a sledování aktuálního i staršího stavu. Střídač posílá data o provozu v pravidelných intervalech do Fronius datového centra. Tato data je možné následně prohlížet přímo na webu či v mobilní aplikaci Solar.web (viz příloha H.14).

Fronius aplikace je možné bezplatně využít k sledování reálného provozu střídače a pro přehled o vyrobené energii a její spotřebě v souhrnu za delší období. V práci jsem

<sup>1</sup> Tím se liší od UPS, která přepne na chod z baterií okamžitě po výpadku.

s jejich pomocí vygeneroval několik obrázků. Z Fronius webové aplikace je zachycená sezónnost výroby elektřiny ze slunce a její následné využití je rozkreslené mezi přímo spotřebovanou energií, energií uloženou do akumulátorů a energií dodanou do rozvodné sítě dle let v příloze D a za celou dobu provozu na obrázku C.5 v příloze C. Dále jsem aplikaci využil k zobrazení celkově vyrobené elektrické energie dle let na obrázku C.6 a pro zobrazení celkově vyrobené elektrické energie za celou dobu provozu v praktické části na obrázku 4.9. Aktuální tok energie je zachycený na obrázku 3.5. Akumulátory jsou plně nabité, solární panely vyrábí 5,25 kW elektřiny, dům využívá 1,23 kW elektřiny a do sítě je dodáváno 3,78 kW. Na obrázku 4.9 nezachycená „chybějící“ hodnota 240 W<sup>1</sup> je vypočtena a zobrazena v praktické části u vyvinuté aplikace RaZberry Home 4.5.



**Obrázek 3.5.** Tok energie zachycený na Fronius iOS aplikaci

Detailní informace z historie nejsou dostupné bezplatně. Společnost Fronius na jedné straně vyžaduje, aby byl střídač připojený k Internetu a data o provozu společnosti bezplatně odesílal, tj. společnost Fronius má díky tomu k dispozici bezplatně data ze všech instalovaných střídačů, které tento požadavek dodržují. Na druhé straně provozovatelé fotovoltaických elektráren mají z odevzdaných dat k dispozici pouze základní historii a statistiky. Pro plný přístup k bezplatně předaným informacím si musí provozovatelé FVE hradit Premium přístup. Bylo by zajímavé prozkoumat, zda je to správně z pohledu práva. Z mého pohledu se jedná o bezplatné shromažďování dat, které přináší společnosti konkurenční výhodu. Existují veřejně dostupné servery, kam je možné svobodně odevzdávat data a později je také plně využívat. Tuto oblast jsem dále neprobádal. Fronius aplikaci jsem využil k zachycení přehledů zmíněných výše a vydal jsem se cestou ukládání dat do vlastní PostgreSQL databáze (viz praktická část 4.7).

Podcílem práce je nevyužívat pro chod cloudové služby, proto jsou Fronius webové a mobilní aplikace použity pouze k zobrazení přehledů v textu práce, ale pro chod domu nejsou využívány.

### ■ 3.1.5 Rozhraní střídače

Střídač byl při instalaci připojen servisním technikem do interní sítě pomocí Wi-Fi a obdržel IPv4 z volného rozsahu od interního DHCP serveru. Na střídači běží integrovaný HTTP server. Pomocí jeho webového rozhraní dokončil technik konfiguraci střídače. Přes webový server je možné se znalostí servisních přístupových údajů do konfigurace FVE zasahovat. Webový server zobrazuje aktuální stav výroby a spotřeby elektrické energie.

<sup>1</sup>  $5,25 - 3,78 - 1,23 = 0,24$



Přístup střídače do Wi-Fi sítě se kvůli jeho umístění ve sklepě ukázal jako nestabilní. Bylo proto nutné přejít na připojení přes kroucené dvojlinky. Nasadil jsem ZyXEL PowerLine Ethernet zobrazený na obrázku 3.6, přepnul jsem konfiguraci střídače na využívání síťové karty a na DHCP serveru jsem mu přidělil IPv4 192.168.4.8. PowerLine Ethernet se skládá ze dvou kusů. Druhý jsem umístil do pracovny. Rozmístění obout je znázorněno na nákresu domu 4.1 oranžově.



**Obrázek 3.6.** ZyXEL PowerLine Ethernet u střídače Fronius

Pro automatizované čtení údajů střídač poskytuje dle Fronius Solar API V1 specifikace [4] webový server s API a JSON výstupem. Konkrétně je možné zavolat API pomocí tohoto URL

```
http://192.168.4.8/solar_api/v1/GetPowerFlowRealtimeData.fcgi
```

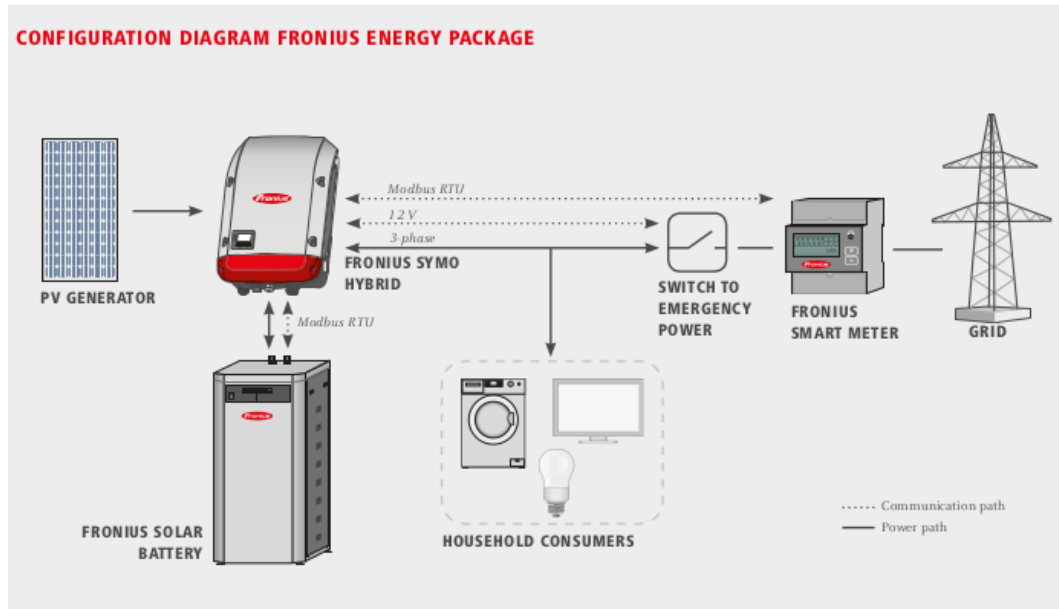
a obdržet v odpovědi informace o aktuálním stavu střídače ve formě indexovaných hodnot v JSON formátu. V tabulce 3.1 jsou vypsány indexy, pomocí kterých je v praktické části přistupováno k hodnotám, které využívá řídicí program. Indexy jsou v tabulce použity ve formě Python JSON zápisu tak, jak jsou následně využívány v aplikaci RaZberry Home 4.5. Pořadí zápisu v tabulce odpovídá pořadí ve výpisu na terminálu a na LCD displeji.

JSON	popis
<code>['Body']['Data']['Inverters']['1']['SOC']</code>	stav nabití akumulátoru, plné nabití = 100%
<code>['Body']['Data']['Site']['rel_Autonomy']</code>	nezávislost na síti, ideální hodnota je 100%
<code>['Body']['Data']['Site']['rel_SelfConsumption']</code>	vlastní spotřeba
<code>['Body']['Data']['Site']['P_PV']</code>	výroba ze solárních panelů $\geq 0$
<code>['Body']['Data']['Site']['P_Load']</code>	průtok střídačem spotřeba (mínus)
<code>['Body']['Data']['Site']['P_Grid']</code>	prodej do sítě $< 0$ , nákup ze sítě $> 0$
<code>['Body']['Data']['Site']['P_Akku']</code>	do (mínus) / z (plus) baterky
<code>['Head']['Timestamp']</code>	časový údaj k načteným hodnotám

**Tabulka 3.1.** Hodnoty použité z Fronius API

### 3.1.6 Zapojení FVE

Zapojení jednotlivých prvků fotovoltaické elektrárny je znázorněno na obrázku 3.7 převzatého z Fronius datového listu [18]. Konkrétní rozmístění jednotlivých částí v rámci domu je zachyceno na nákrese 4.1.



Obrázek 3.7. Diagram zapojení fotovoltaické elektrárny [18]

## 3.2 Počítačová síť

V domě je zprovozněná počítačová síť s privátními IPv4 adresami z třídy C se sítí 192.168.4.0 a routerem na adrese 192.168.4.1. Adresování IPv6 není použito.

Síť LAN je připojena do Internetu pomocí Wi-Fi technologie. Na nákrese domu 4.1 je zobrazena světle zelenými obloučky. Router je PC Engines AMD GX-412TC SOC se třemi síťovými 1000baseT kartami. Operační systém je OpenBSD 6.8 amd64 a vnitřní síť chrání firewallem *pf*. Firewall vychází z nastavení dle [25] a vypadá následovně:

```
doas cat /etc/pf.conf
wired = "em1"
wifi = "athn0"
lyra = "em2"
table <martians> { 0.0.0.0/8 127.0.0.0/8 169.254.0.0/16 \
                  172.16.0.0/12 192.0.0.0/24 192.0.2.0/24 224.0.0.0/3 \
                  192.168.0.0/16 198.18.0.0/15 198.51.100.0/24 \
                  203.0.113.0/24 }

set block-policy drop
set loginterface egress
set skip on lo0

block return # block stateless traffic
pass # establish keep-state
# By default, do not permit remote connections to X11
block return in on ! lo0 proto tcp to port 6000:6010
# Port build user does not need network
block return out log proto {tcp udp} user _pbuild
```

```

match in all scrub (no-df random-id max-mss 1440)
match out on egress inet from !(egress:network) to any nat-to (egress:0)
antispoof quick for { egress $wired $wifi $lyra }
block in quick on egress from <martians> to any
block return out quick on egress from any to <martians>
block all
pass out quick inet
pass in on { $wired $wifi $lyra } inet
pass in on egress inet proto tcp from any to (egress) port { 22 }

```

Router má přidělenou veřejnou statickou IPv4 adresu 109.164.115.11 s přiřazeným doménovým jménem `astaldo.navratil.cz`. Dále je nastaven reverzní záznam z 109.164.115.11 na `astaldo.navratil.cz`. Router umožňuje ssh přístup odkudkoli včetně přístupu z Internetu viz řádek

```

pass in on egress inet proto tcp from any to (egress) port { 22 }

```

v konfiguraci firewallu. V rámci ssh není povoleno přihlašování pomocí hesla. Je nutné použít ssh klíče.

```

grep Password /etc/ssh/sshd_config
PasswordAuthentication no
#PermitEmptyPasswords no

```

Jako typ klíče jsem zvolil dle [26] Ed25519. Na routeru mám uložen tento veřejný ssh klíč

```

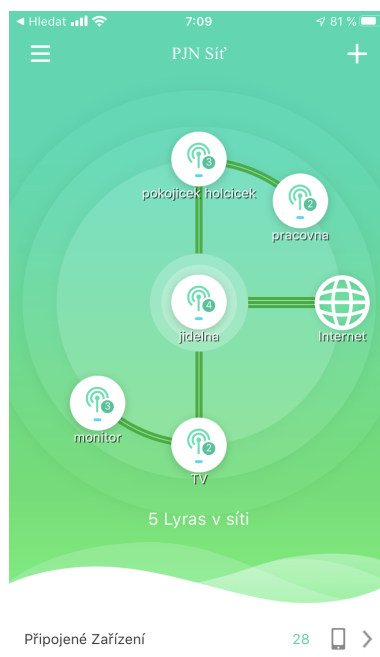
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIM615yS+/5KditOmGtI+1FzAFT0umcI2Qb3
7I+dERVVn jiri@navratil.cz

```

Z vnitřní sítě je router dostupný na adrese 192.168.4.1. Má přiděleno jméno `astaldo` resp. `astaldo.navratil.lan`. Router poskytuje pro LAN služby DHCP, DNS a ARP serveru.

Vrstva síťového rozhraní využívá kroucené dvojlinky, síťový přepínač s gigabitovým Ethernetem, PowerLine Ethernet (na obrázku 4.1 je zakreslen oranžově) a Wi-Fi část.

Dům neměl a nemá rozvedenou strukturovanou kabeláž. Na jeho pokrytí byla využita Wi-Fi technologie. Původní jeden přístupový bod poskytovaný přímo routerem na rozhraní `athn0` nedokázal pokrýt spolehlivě celý dům, a tak byl nahrazen Wi-Fi MESH systémem. Byl nasazen ASUS Lyra AC2200. Na počátku se dvěma satelity. Postupně bylo nutné satelity pro lepší pokrytí přidávat. Nakonec je pokrytí domu zajišťováno pomocí pěti satelitů. V tomto stavu byla Wi-Fi síť při zahájení této práce. Na nákrese domu 4.1 jsou satelity zobrazeny jako světle modré ovály. Jednotlivé satelity jsou pojmenované dle umístění (viz obrázek 3.8 z ASUS Lyra iOS aplikace). Satelity jsou nastaveny do režimu *pouze* přístupového bodu. Routování, DHCP a DNS službu pro ně zajišťuje počítač `astaldo`. ARP záznamy jsou dostupné i na satelitech. V aplikaci výrobce (viz obrázek 3.8) je vidět počet připojených zařízení na jednotlivých satelitech i je možné dohledat jejich MAC adresy. Na obrázku 3.8 je do Wi-Fi sítě připojeno celkem 28 zařízení.



**Obrázek 3.8.** ASUS Lyra AC2200 Wi-Fi MESH systém

Zařízení použitá v diplomové práci, která komunikují na TCP/IP, mají přidělenou konkrétní IPv4 adresu na základě MAC adresy jejich síťového rozhraní viz tabulka 3.2. U Raspberry Pi, která disponují jak Ethernet, tak Wi-Fi rozhraním, je využito Ethernetové rozhraní. V tabulce 3.2 je také uvedena místnost připojení do LAN. V jídelně jsou zařízení zapojena přímo do síťového přepínače. V ostatních místnostech jsou napojena přes kroucenou dvojlinku do Wi-Fi satelitu. U Fronius střídače umístěného ve sklepě je navíc použit PowerLine Ethernet k propojení na síťový přepínač a satelit v pracovně.

IPv4	jméno	místnost	popis
192.168.4.1	astaldo	jídelna	OpenBSD router
192.168.4.7	fronius	sklep	Fronius střídač
192.168.4.9	hub	jídelna	Raspberry Pi se Z-Wave
192.168.4.52	rpi03	pracovna	Raspberry Pi s PostgreSQL
192.168.4.84	rpi19	TV	Raspberry Pi s LCD

**Tabulka 3.2.** Přidělené IPv4 adresy v LAN a umístění zařízení

### 3.3 Raspberry Pi

V domácí počítačové infrastruktuře mám zapojeno devatenáct minipočítačů Raspberry Pi. V této práci ověřím, zda zvládne Raspberry Pi vystupovat v roli řídicího prvku chytré domácnosti.

Dle [8] vytvořil koncepci minipočítače Raspberry Pi kolem roku 2006 doktor Erben Upton se spolupracovníky v počítačové laboratoři Cambridgeské univerzity v Anglii. Vedlo je k tomu přesvědčení, že vytvořením nenákladného počítače určeného k učení a experimentování se podaří zvrátit negativní trend ve znalostech a dovednostech studentů přicházejících studovat počítačovou vědu. Založili nadaci Raspberry Pi Foundation na podporu počítačové vzdělanosti. Dle [19] byla nadace zaregistrována 5. května 2009 a k 31. prosinci 2019 měla ve vedení 10 členů, dále 184 zaměstnanců a pomáhalo ji

v té době 17 241 dobrovolníků. Sídlí na adrese 37 Hills Road, Cambridge, CB2 1NT, UK a oficiální webová adresa je <https://www.raspberrypi.org/>. Dle [20] bylo na konci roku 2019 prodáno přes 30 milionů kusů Raspberry Pi.

Jak jsem uvedl výše, já jsem přispěl do tohoto počtu mými devatenácti kusy. Pro použití výhradně v této práci jsem z nich vybral dva Raspberry Pi minipočítače. U obou vybraných deskových počítačů se jedná o stejný model. Konfigurace je následující:

```
tail -n 14 /proc/cpuinfo | grep -Ev "Serial|Features"
processor          : 3
model name        : ARMv7 Processor rev 4 (v7l)
BogoMIPS          : 51.20
CPU implementer   : 0x41
CPU architecture : 7
CPU variant       : 0x0
CPU part          : 0xd03
CPU revision      : 4

Hardware          : BCM2835
Revision          : a02082
Model             : Raspberry Pi 3 Model B Rev 1.2
```

Jedná se o modely nesoucí označení Raspberry Pi 3 Model B Rev 1.2. Dle revize a02082 a dokumentu [21] se jedná o modely vyrobené u SONY v UK a jejich kapacita operační paměti je 1GB. To prokazuje i tento výpis přímo z počítače:

```
dmesg | grep Memory: | cut -c-48
[    0.000000] Memory: 880472K/970752K available
```

Zajímavostí je, že položka „Hardware: BCM2835 z /proc/cpuinfo“ je chybná viz. [21]. Ve skutečnosti se jedná o SoC Broadcom BCM2837, což dokazuje i následující výpis:

```
lscpu | grep -v Flags
Architecture:      armv7l
Byte Order:        Little Endian
CPU(s):            4
On-line CPU(s) list: 0-3
Thread(s) per core: 1
Core(s) per socket: 4
Socket(s):         1
Vendor ID:         ARM
Model:             4
Model name:        Cortex-A53
Stepping:          r0p4
CPU max MHz:       1200,0000
CPU min MHz:       600,0000
BogoMIPS:          38.40
```

Cortex-A53 je 64bitový ARM procesor. V Raspberry Pi 3 Model B Rev 1.2 může být taktován maximálně na frekvenci 1.2 GHz. Dle výpisu výše se frekvence tohoto minipočítače může pohybovat od 600 MHz do 1200 MHz. Frekvence se průběžně mění v závislosti na zátěži. Dle aktuální frekvence se vypočítává hodnota BogoMIPS. Ve výpisech výše je u /proc/cpuinfo 51.20 BogoMIPS, u lscpu 38.40 BogoMIPS a během psaní těchto řádků jsem uměle minipočítač zatížil pomocí příkazu `diff`<sup>1</sup> a hodnota BogoMIPS

<sup>1</sup> Nechal jsem porovnat dva 380M velké textové soubory.

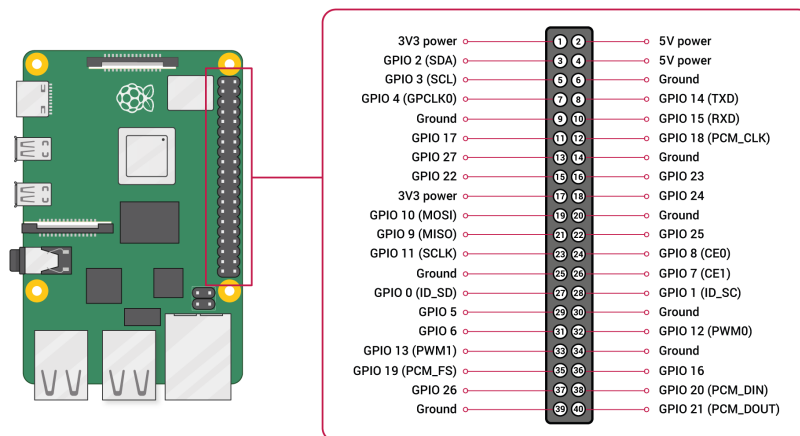
vzrostla na 76.80 BogoMIPS. V praktické části bude ukázané, zda je tento rozsah, tedy výkon CPU, pro potřeby práce dostatečný.

Dle specifikace [28] je Raspberry Pi 3 Model B Rev 1.2 nejmladším modelem ze třetí generace deskových počítačů Raspberry Pi. Mezi jeho parametry patří:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 Wi-Fi a Bluetooth Low Energy na desce
- 100 Base Ethernet
- 40pinové GPIO
- 4x USB 2 port
- HDMI
- Micro SD slot
- napájení přes Micro USB do 2.5A

Při volbě konkrétních kusů jsem nejprve zvolil *rpi04.navratil.lan* s IPv4 192.168.4.9 a osadil ho přes GPIO rozhraní Z-Wave modulem RaZberry. Propojení minipočítače Raspberry Pi s RaZberry je vidět na obrázku 3.12. O technologii Z-Wave píše v následující podkapitole 3.4 a další informace k RaZberry jsou uvedeny v části 3.5.1. První instalace a následující aktualizace až po aktuální verzi jsou popsány v praktické části 4.6.

Následně jsem také nasadil *rpi19.navratil.lan* s IPv4 192.168.4.83, kvůli potřebě volného GPIO rozhraní pro LCD displej. Zapojení LCD displeje na použitém Raspberry Pi 3 Model B Rev 1.2 je vidět na fotografii 4.8. Nákres zapojení a popis GPIO pinů u Raspberry Pi 4<sup>1</sup> je znázorněno na obrázku 3.9 převzatého z [27]).



Obrázek 3.9. Nákres a popis GPIO rozhraní [27]

První stroj osazený rozšířením pro chytrou domácnost je centrálním bodem pro jednotlivé Z-Wave prvky domácnosti. Druhý stroj osazený LCD displejem slouží k běhu řídicího programu vytvořeného v rámci této práce a prostřednictvím prvního stroje kontroluje Z-Wave síť. Pro řídicí program jsem zvolil název RaZberry Home a používám pro něho dvoupísmenné označení *rh*. Aplikace běžela původně na prvním stroji, ale kvůli rozšíření řešení o malý LCD displej s GPIO napojením jsem potřeboval použít další Raspberry Pi, neboť na prvním stroji bylo GPIO již obsazené RaZberry modulem.

<sup>1</sup> Umístění a zapojení GPIO na Raspberry 4 je identické s použitým Raspberry Pi 3 Model B Rev 1.2.

Na minipočítače Raspberry Pi je možné instalovat různé verze operačních systémů určené pro ARM platformu. Raspberry Pi Foundation [19] poskytuje jako doporučovaný a pro Raspberry Pi optimalizovaný operační systém Raspberry Pi OS vycházející z GNU Debian Linuxu. V době započetí práce se jmenoval Raspbian a byla k dispozici verze 8 se jménem Jessie. V březnu 2021 před odevzdáním práce je aktuální verze 10 se jménem Buster a na minipočítačích Raspberry Pi jsou nainstalované tyto verze Raspberry Pi OS operačního systému:

```
navratil@rpi04:~ $ lsb_release -d
Description:    Raspbian GNU/Linux 9.13 (stretch)
navratil@rpi19:~ $ lsb_release -d
Description:    Raspbian GNU/Linux 10 (buster)
```

Kroky vedoucí k tomuto stavu a vysvětlení, proč není `rpi04 (hub)` na poslední verzi operačního systému Raspberry Pi OS jsou popsány v praktické části.

### 3.4 Chytrá domácnost

Domácnost provozující fotovoltaickou elektrárnu má již díky střídači přidanou chytrost. Úlohou střídače je řídit tok vyrobené energie, spotřebu energie i odběr ze sítě či dodávky do sítě. Z vyrobené energie přednostně zásobuje domácnost, následně ji ukládá do baterií a teprve pokud není možné vyrobenou energii využít, tak ji odevzdává<sup>1</sup> do veřejné sítě.

Členové domácnosti mohou na webovém rozhraní střídače nebo ve Fronius iOS aplikaci pozorovat aktuální tok elektřiny, tj. kolik elektrické energie je vyráběno solárními panely, jaká je její spotřeba, stav nabití akumulátorů a také, zda je elektrická energie dokupovaná ze sítě či je do ní naopak posílána. Uživatelé FVE se postupně bez potřeby sledování střídače naučí poznávat, jaký vliv má na výrobu roční období, denní doba i například hustota mraků<sup>2</sup> a přizpůsobovat tomu cíleně dobu zapínání spotřebičů s větší spotřebou elektrické energie jako varná konvice, myčka nádobí, pračka, sušička, vysavač, pečící trouba, mikrovlnná trouba, kávomat, topinkovač, ...

U spotřebičů vyžadujících nepřetržitý provoz jako router, servery, Raspberry Pi počítače, lednička a mraznička do provozu s cílem uspořit uživatelé nezasahují. U těchto zařízení přichází na řadu jiná chytrost či spíše schopnost domácnosti, a to ostrovní režim. FVE zvládne případný výpadek dodávky ze sítě nahradit vyráběnou elektřinou ze solárních panelů a z uložené energie v akumulátorech.

Již po prvních dnech provozu FVE bylo zřejmé, že výkon solárních panelů a kapacita akumulátorů jsou dostatečné k tomu, aby ve dnech s večer plně nabitou baterií stačila pro noční a brzký ranní provoz domu energie uložená v akumulátorech a následná spotřeba se již pokryje ze slunečního záření. V ideální situaci je tedy možné dosáhnout 100% soběstačnosti, což se také opakovaně dělo. K tomu, aby se dosahovalo 100% soběstačnosti co nejčastěji, bylo vhodné přidat další úroveň *chytrosti domácnosti*. U spotřebičů, které mají velkou spotřebu a nemusí běžet v noci případně mohou běžet dokonce jen pokud je k dispozici dostatečné množství vlastní elektrické energie, by měla tato další úroveň chytrosti přidat automatizované na členech domácnosti nezávislé zapínání a vypínání těchto spotřebičů. Přidání další úrovně chytrosti se věnuje tato práce a nasazuje k tomu dříve zmíněný Raspberry Pi minipočítač.

100%  
nezávislost

<sup>1</sup> Sloveso odevzdává je použito záměrně. Společnost PRE elektřinu nevykupuje.

<sup>2</sup> Věděli jste, že při mírném dešti může FVE vyrábět více než při hustých mracích bez deště?



**Obrázek 3.10.** Střídač, akumulátory a odvlhčovače ve sklepě

Jako vhodné spotřebiče, které mohou být Raspberry Pi minipočítačem řízeny, se nabízí odvlhčovače, 80l elektrický bojler a elektrický kotel ústředního topení. Bojler a odvlhčovače jsou v celoročním provozu. Elektrický kotel na obrázku 3.4 je v chodu pouze v topné sezóně. Odvlhčovače jsou v domě čtyři. Dva jsou umístěny ve sklepě viz obrázek 3.10, jeden v pracovně<sup>1</sup> a jeden v koupelně.

K ovládání a monitorování stavu spotřebičů bude potřeba dům doplnit o tyto komponenty:

- řídicí jednotka - rozšíření Raspberry Pi o schopnost řídit chytré komponenty
- chytrá zásuvka - zapínání a vypínání odvlhčovače
- chytrý 230V spínač - zapínání a vypínání elektrického kotle a elektrického bojleru
- chytrý termostat k radiátoru - zjišťování teploty, zapínání a vypínání radiátoru

Na předchozích řádcích jsem vybral vhodné spotřebiče k řízení spotřeby. Současně s nimi jsem i vypsál, jaké prvky chytré domácnosti je nutné zajistit. Teď je potřeba najít technologii, na které budou tyto prvky chytré domácnosti postaveny.

Dle [1] je nutné nejprve zvolit, jak budou prvky chytré domácnosti propojeny a nabízí tyto způsoby:

- drátové propojení (například BACNet nebo Instabus) - vyžaduje položení kabelů při stavbě či rekonstrukci
- PowerLine - využívá 230V rozvod (obdobu PowerLine Ethernetu v podkapitole 3.2)
- bezdrátové řešení - několik variant, nasazení bez zásahů do domu, cenově dostupné

V mé situaci, kdy je již dům postavený, rekonstrukci neplánuji a pro LAN jsem zvolil Wi-Fi místo dodatečného zavádění strukturované kabeláže, budu i pro nasazení chytré domácnosti hledat bezdrátové řešení.

V rámci magisterského studia jsem se setkal s těmito bezdrátovými technologiemi: Wi-Fi, Bluetooth, Zigbee a Z-Wave. Pro kvalitní pokrytí domu bude vhodné nasadit nějakou formu MESH systému. Raspberry Pi podporuje Wi-Fi a Bluetooth Low Energy. Pro Wi-Fi už v domě vybudovaný MESH systém je, ale Wi-Fi má oproti ostatním třem technologiím vyšší spotřebu a nehodí se tudíž do zařízení napájených z monočlánků. Napájení z monočlánků bude nutné použít u termostatů. Bluetooth Low Energy podporuje MESH systém, má nízkou spotřebu, ale nepodařilo se mi najít potřebná koncová

<sup>1</sup> Jedná se o technickou místnost s počítači. Příznačnější označení by bylo serverovna.



zařízení. Zbývající dvě technologie Zigbee a Z-Wave podporují MESH systém, mají nízkou spotřebu, podporují šifrování a existují k nim koncová zařízení, která potřebují: chytrá zásuvka, 220V přepínač a chytrý termostat. Pro výběr mezi Zigbee a Z-Wave mi pomohlo nalezení Z-Wave řídicí jednotky připojitelné na Raspberry Pi přes GPIO. Rozhraní GPIO jsem používal s Adafruit USB to TTL zařízením [29] pro napájení a ovládání Raspberry Pi přes sériovou konzoli a testování OpenBSD na Raspberry Pi. Použití GPIO bylo stabilní a bezproblémové. Naopak u USB zařízení jsem v minulosti na platformě x86 narážel na problémy s ovladači, nastavením v BIOSu a konflikty mezi více zařízeními. Na platformě ARM jsem doposud zkušenosti s USB zařízeními neměl. Subjektivně vypadá Raspberry Pi se Z-Wave modulem na GPIO kompaktněji a lépe, než se zastrčeným modulem do USB portu. Objektivně při použití GPIO nepotřebuje Raspberry Pi větší plochu a zejména nehrozí nechtěné odpojení USB zařízení. V okamžiku rozhodování mi tyto rozdíly výše stačily pro učinění rozhodnutí. V té době jsem ještě nepsal diplomovou práci. Nyní bych použil multikriteriální analýzu, aby byla volba lépe podložena a odůvodnitelná. V březnu 2021 před odevzdáním práce se situace oproti roku 2017 změnila a volba by nejspíše dopadla jinak viz dále v kapitole 5.

## 3.5 Z-Wave

Technologie Z-Wave resp. její novější verze označovaná jako Z-Wave Plus nebo někdy také zkráceně jako Z-Wave+ [30] podle [1] splňuje požadavky na spolehlivost, bezpečnost a interoperabilitu.



Obrázek 3.11. Z-Wave Plus logo [30]

Na německém Amazonu jsem našel všechny potřebné komponenty včetně termostatů od německého výrobce EUROTRONIC Technology GmbH. Objednal jsem tato Z-Wave zařízení

- řídicí jednotka - Z-Wave.Me RaZberry 2 modul ZMEERAZ2 á 59,98 €<sup>1</sup>
- chytrá zásuvka - FIBARO Wall Plug FGWPF-102-5 á 50,72 €<sup>2</sup>
- chytrý 230V spínač - FIBARO Double Switch 2 FGS-223 á 49,99 €<sup>3</sup>
- chytrý termostat k radiátoru - EUROTRONIC Spirit Z-Wave Plus á 50,60 €<sup>4</sup>

a Amazon mi je bezplatně doručil až domů. V první objednávce jsem nakoupil po jednom kuse na vyzkoušení. V praktické části je popsána jejich instalace, testování a následné dokoupení dalších kusů a pokrytí celého domu.

### 3.5.1 RaZberry

Řídicí Z-Wave jednotka určená pro připojení na Raspberry Pi rozhraní GPIO je výrobcem Z-Wave.Me pojmenovaná jako RaZberry [31]. Umožňuje připojení na všechny čtyři

<sup>1</sup> ASIN B01LVV6GE4

<sup>2</sup> ASIN B01HM0MQ8O

<sup>3</sup> ASIN B01HM0MQAW

<sup>4</sup> ASIN B075X257NC

generace<sup>1</sup> minipočítače Raspberry Pi. Bylo ji tedy možné použít na mém Raspberry Pi 3 Model B Rev 1.2.

Při objednání jsem zvolil model ZMEERAZ2 [31] s frekvencemi kolem 868 MHz, který umožňuje nastavení EU platné i pro Českou republiku a variantu s větším obvodem kvůli větší anténě a tudíž delšímu dosahu.

RaZberry je přes GPIO konektor napájen a probíhá přes něj komunikace.

Na obrázku 3.12 vpravo dole je vidět napojení RaZberry na Raspberry Pi za použití deseti pinů GPIO rozhraní. Vpravo na boku je zasunuta microSD karta. Vpravo nahoře je přivedeno napájení do microUSB portu. Vlevo nahoře je do RJ-45 přiveden kabel s kroucenými dvojlinkami. Tento kabel propojuje Raspberry Pi se switchem.



Obrázek 3.12. Z-Wave modul RaZberry na Raspberry Pi

## 3.6 Testování

Pro testování jsem zvolil inkrementální techniku z agilní metodiky s přímým nasazováním do reálného provozu. Díky této technice bude možné posoudit výsledek rozšiřování či oprav kódu ihned či krátce po nasazení. V doslova domácím prostředí jsem schopen nasazení změn načasovat tak, aby případné chyby neměly zásadní dopad na chod domácnosti.

### 3.6.1 Persony

Jako *persony* použiji:

- děti - žijí v domě, nechtějí být omezovány řešením chodu domácnosti, v topné sezóně chtějí mít v pokojíčkách a v koupelně teplo
- žena - nechce být omezována řešením chodu domácnosti, v topné sezóně chce mít v domě o 1 °C tepleji než děti
- vývojář - chce co nejvíce automatizovat chod domácnosti, co nejvíce ušetřit za elektrickou energii a co nejvíce ji spotřebovat oproti odevzdání do sítě, změny financuje, je zodpovědný za volbu hardware i software, navrhuje a implementuje automatizaci, změny nasazuje a testuje, požadavky dětí a ženy přidává ke svým vlastním

<sup>1</sup> Před odevzdáním této práce je již dostupná čtvrtá generace minipočítače Raspberry Pi.

### ■ 3.6.2 Metoda

Pro testování nasadím metodu „Behavior Tests“ popsanou v [32]. Dle [32] se tento typ testování zaměřuje na individuální vlastnosti aplikace nebo celého systému a na chování, které musí demonstrovat jako splnění požadované vlastnosti. Metoda tedy testuje chování. Na popis mnou testovaného chování využiji DSL Gherkin [33]. Nebudu připravovat automatizované testování. Chování budu testovat ručně.

Například zápis v DSL Gherkin pro testování ovládání elektrického kotle v topné sezóně na základě pouze dne v týdnu a denní doby je následující:

**Požadavek:** Ovládání elektrického kotle dle denní doby

Zapínání a vypínání elektrického kotle  
dle dne v týdnu a denní doby.

**Scénář:** Elektrický kotel je zapnutý

Pokud je topná sezóna  
A dnes se má topit  
A v tento čas se má topit  
Pak je elektrický kotel zapnutý

**Scénář:** Elektrický kotel je vypnutý

Pokud je topná sezóna  
A dnes se má topit  
Ale v tento čas se nemá topit  
Pak je elektrický kotel vypnutý

## ■ 3.7 Programovací jazyk

Teoretická východiska, existující infrastruktura, znalosti a zkušenosti ze studia, ze zaměstnání i z projektů realizovaných jako živnostník mne nasměrovaly na programovací jazyk Python. Pro inkrementální techniku z agilní metodiky Python vhodný je. Pro HTTP požadavky a zpracování JSON jsou dostupné moduly `requests` a `json`. Stejně tak existují moduly pro přístup do relační databáze PostgreSQL `psycopg2` a pro ukládání časových řad do InfluxDB `influxdb`.

Při volbě verze jsem vybral Python 3, neboť verze Python 2 již měla ohlášený konec podpory na 1. ledna 2020.

V rámci vývoje i v rámci provozu použiji virtuální prostředí `venv`.

Pro dokumentaci kódu a jeho použití využiji modul `mkddocs` a pro kontrolu stylu psaní Python kódu modul `pycodestyle`.

## ■ 3.8 Verzovací systém

Zdrojový kód je verzován a ukládán na můj Git server do vyčleněného repozitáře `rh.git`. Velikost repozitáře k 26. březnu 2021 je 7,2 MB viz 3.3. Do stejného repozitáře je ukládána také dokumentace kódu ve formě MkDocs.

```
doas du -sh /home/git/rh.git
7.2M /home/git/rh.git
```

**Tabulka 3.3.** Velikost Git repozitáře se zdrojovým kódem a jeho dokumentací v MkDocs.

Diplomová práce je verzována v decentralizovaném verzovacím systému *Fossil SCM* a jako repozitář používám `CZU_PEF.fs1` obsahující data za celou dobu mého studia na ČZU PEF. Velikost repozitáře k 26. březnu 2021 je 4,1 GB. V lokální síti mám synchronované kopie tohoto repozitáře na notebooku, na serveru a na Raspberry Pi 4 se dvěma monitory.

## 3.9 Dokumentace

Během přípravy i v průběhu realizace jsem si užíval psaní plnicím perem na A4 papíry s nízkou gramáží<sup>1</sup> a do diářů<sup>2</sup> metodou *Bullet Journal*. Používal jsem hlasový záznam na diktafon, do telefonu a na počítači. Stejně tak jsem používal elektronickou formu zápisu do textových souborů s Markdown či plain  $\text{\TeX}$  formátováním.

Do této práce jsem vybral a doplnil části realizující cíl a podcíl práce a dále části, které považuji něčím důležité či alespoň zajímavé. Jak jsem psal v úvodu 1, kód i dokumentaci nadále vylepšuji a mám v úmyslu v tom pokračovat i po odvezdání práce.

Zdrojový text diplomové práce byl napsán v textovém editoru Vim a vysázen v plain  $\text{\TeX}$ u [38] Donalda Knutha [37], využívá rozšiřující makra *OPmac* [39] a šablonu *CTUstyle* pro sazbu studentských závěrečných prací RNDr. Petra Olšáka [39]. Šablonu jsem upravil dle grafického manuálu ČZU [23]. Novou šablonu pojmenoval *CULSstyle* a připravil pro ni repozitář na GitHubu <https://github.com/plavcik/CULSstyle>. Plánuji po státních závěrečných zkouškách svoji lokální verzi *CULSstyle* upravit a zveřejnit na GitHubu.

Struktura diplomové práce je nastavena dle ČZU brožury „Požadavky pro předkládání diplomových prací“ [24].

Ke generování PDF souboru je využít příkaz `make` s pravidly definovanými v souboru `Makefile` viz 3.4.

```
filename=diplomova_prace

${filename}.pdf: *.tex
    pdfcsplain ${filename}.tex

tar: ${filename}.pdf
    tar czf ${filename}-'date +%Y%m%d'-'whoami'.tgz *.tex images/

bib:
    pdfcsplain ${filename}
    pdfcsplain ${filename}

read:
    xpdf -z page -ps ${filename}.ps -paper A4 -eol unix ${filename}.pdf &

clean:
    rm -i ${filename}.*{pdf,ps,log,dvi,lg,idv,html,ref,txt}
```

**Tabulka 3.4.** Makefile pro generování PDF souboru s diplomovou prací.

O verzování diplomové práce píše v podkapitole 3.8.

<sup>1</sup> 60g/cm<sup>2</sup> a 70g/cm<sup>2</sup>

<sup>2</sup> diář *Bullet Journal* LEUCHTTURM 1917

Dokumentace aplikace je součástí zdrojového kódu. Používám Markdown soubory a MkDocs [36] dokumentaci.

## 3.10 Databáze

### 3.10.1 Časové řady

Data ze Z-Wave zařízení budou průběžně ukládána do time-series databáze a dotazy z řídicí aplikace budou směřovány do této databáze. Díky tomu se odlehčí provozu Z-Wave sítě a také dojde k prodloužení životnosti monočlánků v termostatech.

Software Z-Way běžící na řídicí jednotce RaZberry obsahuje rozšíření na podporu InfluxDB. V praktické části bude nainstalována a nakonfigurována tato databáze.

### 3.10.2 Relační databáze

Pro sledování maximálních hodnot a pro uchování kompletních JSON dat ze střídače bude nasazen relační databázový systém. V domě je nainstalován RDBMS PostgreSQL, a tak bude vytvořena a využita databáze pro účely této práce v PostgreSQL.

## 3.11 Hesla

V této podkapitole se chci zmínit o tom, jakým způsobem používám hesla u uživatelských účtů a u aplikačních účtů.

V diplomové práci na místech určených pro heslo nepíši reálně použité heslo ani tam nedávám hvězdičky, ale používám slovo `heslo`. V textu práce sice prozrazuji topologii sítě, MAC adresy, IPv4 adresy, operační systém použitý na routeru, nastavení firewallu, jeho veřejnou statickou IPv4 adresu a spoustu dalších informací. V odevzdané příloze `rh-src_and_mkdocs.tar` jsem dokonce ponechal konfigurační údaje v `config.py` včetně hesel. Přesto jsem se nakonec rozhodl alespoň v textu práce reálná hesla nahradit slovem `heslo`. Jsem si vědom, že by to mohlo působit jako nekonzistence. Dovoluji si proto ještě doplnit, jakým způsobem k heslům přistupuji.

V rámci domácnosti mám na strojích Raspberry Pi hesla pro účty uživatele `root` a pro uživatele `navratil` generovaná. Původí účty `pi` jsou smazané a jejich `sudo` práva odebrána. Hesla jsou pro každý minipočítač unikátní. Pro práci skriptů vytvářím speciální uživatele jako například `script01`. Pro přihlašování na Raspberry Pi počítače používám výhradně veřejné `ed25519` ssh klíče a přihlašování pomocí hesel přes `ssh` nemám povolené. Privátní `ed25519` ssh klíče mám pouze na počítačích, které mají šifrovaný disk. Pro privátní klíče používám netriviální, ale zapamatovatelné heslo.

Pro generování hesel používám tento příkaz.

```
openssl rand -base64 2000 | cut -c 1-20 | head
VLjWmyV/K5lZvs3mwvNZ
pHshClJD/zt59rHcD1tx
OsrpAEPblEUXBIKSe1Gi
XqnFA5vt9lxUY7GWiql7
8bCHuwVoFuCNtM11RgqA
rpMG+UUPP1KDZeU5ikY9
eIVuEPCWPInjTYR4cFnq
AlCOhi4DxPrtaA5fzCWz
wDddaooeRQz64JZED0s4
v5oPOCToGBB5PaaexL8A
```

# Kapitola 4

## Praktická část

Praktická část začala v roce 2017. Fotovoltaická elektrárna byla spuštěna do provozu 21. dubna 2017 ve 12:39. Její instalaci a spuštění realizovala externí firma. V této kapitole se dočtete, jakým způsobem se domácnost postupně měnila díky nasazení minipočítače Raspberry Pi a prvků chytré domácnosti. Realizace cíle a podcíle práce je zde popisována slovně s ukázkami ze zdrojového kódu, z instalačních logů a ukázek aktuální konfigurace počítačů. Kompletní zdrojový kód včetně provozní dokumentace je součástí zvlášť odevzdané přílohy.

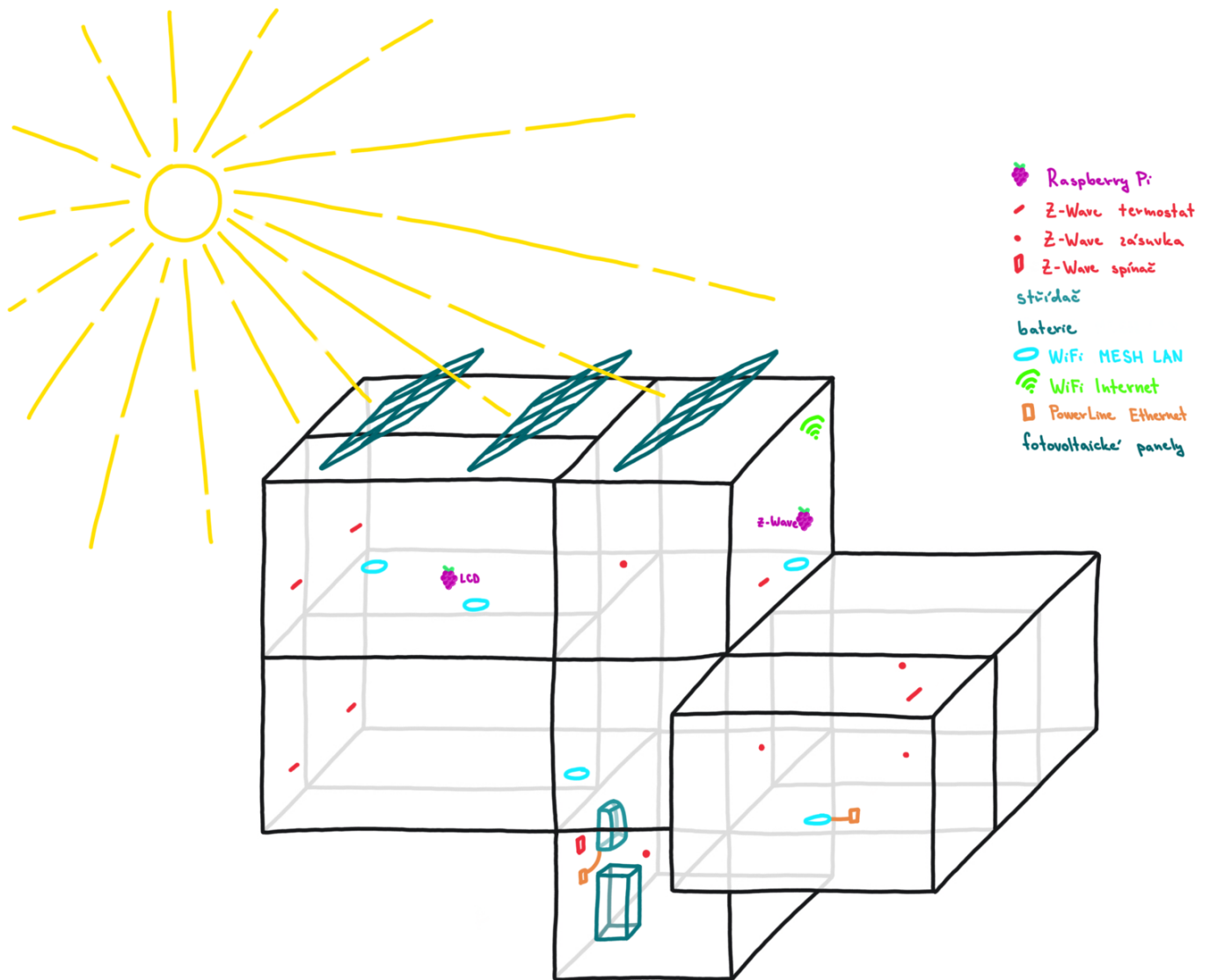
V rámci realizace jsem střídal role jako investor, architekt, vývojář, tester a interní zákazník. Z pohledu definovaných person jsem vystupoval v roli „vývojář“. Bylo užitečné, že jsem díky inkrementálnímu nasazování viděl „ihned“ výsledky a mohl případně nasazené změny opravit či dále vylepšovat. Cenná pro mne byla zpětná vazba od dětí, které se mnou v domě žijí. Hledání odpovědí na jejich nikdy nekončící otázky bylo někdy náročné, ale pro moji práci rozhodně přínosné. Teprve v okamžiku, kdy jsem byl schopen zvědavost dětí uspokojit, byla moje odpověď dostatečně zjednodušená a jasně vysvětlená.

Většina následující podkapitol popisuje změny, kterými dům v souvislosti s touto prací prošel. Pořadí kapitol odpovídá pořadí, v jakém byly změny implementovány. Pro rychlé nalezení hlavní části řešící cíl a podcíl práce je možné přejít přímo na podkapitolu 4.5.

### 4.1 Nákres domu

Pro představu rozložení zařízení v rámci domu jsou na obrázku 4.1 zakreslena použítá zařízení a jejich přibližné umístění v domě. Fialovou barvou a symbolem maliny jsou zakresleny Raspberry Pi počítače, červenou barvou Z-Wave koncová zařízení jako termostaty na radiátorech, zásuvky a spínač elektrického kotle ústředního tepení ve sklepě. Zelenomodrou barvou jsou vyobrazeny součásti fotovoltaické elektrárny jako solární panely na rovné střeše, akumulátory a střídač ve sklepě. Světle modře je vidět vnitřní počítačová síť tvořená pěti satelity ASUS Lyra AC2200 Wi-Fi MESH systému. Pomocí ZyXEL PowerLine Ethernetu nakresleného oranžově je do počítačové sítě napojen střídač. Celý dům je do Internetu připojen pomocí Wi-Fi zakreslené světle zelenými obloučky.

Nákres zachycuje stav řešení k 13. prosinci 2020. Následující řádky popisují kroky, které k tomu stavu vedly.



Obrázek 4.1. Náskres domu s umístěním Z-Wave, FVE a dalších zařízení.

## 4.2 Instalace řídicí jednotky

Tato podkapitola popisuje instalaci Z-Wave řídicí jednotky RaZberry na minipočítač Raspberry Pi. Pro instrukce jsem použil Z-Way User and Programmer Manual [5].

Po obdržení první sady Z-Wave zařízení z Amazonu viz 3.5 jsem začal s instalací řídicí jednotky Z-Wave.Me RaZberry 2 modul ZMEERAZ2. RaZberry jsem připojil na Raspberry Pi `rpi04.navratil.lan` pomocí GPIO rozhraní. Řídicí jednotka využívá krajních 10 pinů GPIO rozhraní Raspberry Pi. Přes toto rozhraní je napájena a probíhá přes něj také komunikace. Kroucenou dvojlinkou jsem propojil přes RJ-45 konektor Raspberry Pi se switchem. Počítač jsem zatím nezapojoval do napájení. Pustil jsem se do přípravy paměťové karty. Zakoupil jsem SanDisk Ultra microSDHC 128 GB A1 Class 10 UHS-1.

Řídicí software pro RaZberry se jmenuje Z-Way. Je možné ho instalovat na již běžící Raspbian systém, nebo použít obraz disku s předinstalovaným operačním systémem i softwarem Z-Way. Obraz disku je určený k binárnímu nakopírování na paměťovou kartu. Vzhledem k mým doposud žádným zkušenostem se Z-Way

jsem se rozhodl využít obraz disku. Odkaz na soubor ke stažení je k dispozici na <https://z-wave.me/z-way/download-z-way/>. Já jsem použil pro stažení obrazu disku toto URL <https://razberry.z-wave.me/z-way-server/razberry.img.zip>. Stažený soubor `razberry.img.zip` měl SHA256 kontrolní součet `11a610d5b0d557fa971b097fc910a4d1cf7d8cfd24075cd10c7c96410f6a0323` a po rozbalení jsem získal soubor `razberry-2.2.5-wifi_public.img` s SHA256 kontrolním součtem `e0baae93aa31e8c97cc17651fd5ad481dd13b7a3948eb21752caf571698cd0a9`. Tento soubor jsem na počítači s OpenBSD nakopíroval na paměťovou kartu a následně ověřil bezchybnost nakopírování pomocí těchto příkazů:

```
doas dd if=razberry-2.2.5-wifi_public.img of=/dev/rsd9c bs=32k
doas cmp razberry-2.2.5-wifi_public.img /dev/rsd9c
```

Paměťovou kartu s nakopírovaným obrazem disku jsem vložil do microSD slotu v Raspberry Pi, minipočítač jsem v obyvaku připojil prostřednictvím HDMI kabelu na televizi, přes RJ-45 jsem napojil Raspberry Pi na switch u televize, do USB jsem zasunul USB modul k bezdrátové klávesnici Logitech K400 Plus a připojil microUSB napájecí konektor do zdroje.

Při bootování z paměťové karty došlo nejprve ke zvětšení Linuxové partition a na ní umístěném filesystému ext4 na maximální možnou velikost 119 GB. Partition určená pro bootování s vfat filesystémem zůstala beze změny na velikosti 63 MB. Po rebootu se Z-Way automaticky nakonfiguroval. Součástí RaZberry je i licence pro používání Z-Way programového vybavení. V rámci instalace byla tato licence použita a pro moji instalaci bylo přiděleno ID 99065. Zvolil jsem si heslo pro administrátorský účet a pro přístup do cloudového řešení Z-Way Remote Access na adrese <https://find.z-wave.me/> a do lokálního rozhraní k Z-Way serveru na <http://hub:8083/>. Pro počítač `rpi04.navratil.lan` s RaZberry a Z-Way serverem jsem nastavil na všech počítačích, ze kterých jsem na něj chtěl přistupovat, do souboru `/etc/hosts` alias `hub`. Konkrétně jsem přidal řádek `192.168.4.9 rpi04 rpi04.navratil.lan hub`.

Na webovém rozhraní Z-Way serveru je možné přidávat Z-Wave zařízení, přiřazovat je do místností a vkládat na úvodní přehledovou stránku, kontrolovat stav baterií, zálohovat konfiguraci a dělat další úkony. Pro přihlášení je nutné použít jméno `admin` a heslo zvolené při instalaci.

Po dokončení instalace jsem minipočítač Raspberry Pi vypnul a přesunul do jídelny. V jídelně jsem ho připojil přes RJ-45 na switch a zapnul zapojením napájení do IKEA KOPPLA 3cestné USB nabíječky. Při následném používání a monitorování stavu jsem z výpisu příkazu `sudo dmesg` zjistil, že sdílení USB nabíječky s dalšími zařízeními<sup>1</sup> snižuje její výkon a Raspberry Pi si na tuto situaci stěžuje.

```
rpi_firmware_get_throttled: 10 callbacks suppressed
Under-voltage detected! (0x00050005)
Voltage normalised (0x00000000)
```

Situaci jsem vyřešil zakoupením originálního Raspberry Pi zdroje a zapojením `rpi04.navratil.lan` na něj.

## 4.3 Instalace Z-Wave zařízení

Po nainstalování a nakonfigurování RaZberry jako centrální řídicí jednotky pro Z-Wave síť je nyní na řadě vytváření této sítě přidáváním Z-Wave zařízení.

<sup>1</sup> Nabíječku jsme používali k dobíjení mobilních telefonů.




Pro přidání nového Z-Wave zařízení se použije <http://hub:8083/smarthome/#/devices> → Add New → Add new Z-Wave Device and identify it automatically<sup>1</sup> a na poslední stránce před spuštěním procesu přidání je vhodné zvolit Secure pro zabezpečené připojení u zařízení podporující Z-Wave Plus a případně Reset Device pro přepnutí zařízení před přidáním do továrního nastavení<sup>2</sup>. Tlačítkem Start se spustí proces přidávání. Do režimu přidávání nového zařízení je také možno přejít přímo použitím <http://hub:8083/smarthome/#/zwave/inclusion>.

Zařízení dostávají svoje Device ID sekvenčně z množiny přirozených čísel. Řídící modul RaZberry má číslo 1 a detaily o něm je možné zobrazit na <http://hub:8083/expert/#/configuration/interview/1>. Termostat EUROTRONIC Spirit Z-Wave Plus, který byl namontován k radiátoru v obýváku hned jako první následující zařízení po RaZberry, dostal v roce 2017 při první instalaci číslo 2, ale v březnu 2021 má číslo 35 a je dostupný na <http://hub:8083/expert/#/configuration/interview/35>. K přesunutí od Device ID 2 až k současnému Device ID 35 docházelo postupně. Tato situace nastávala, když jsem včas nevyměnil v termostatu monočlánek resp. pokud jsem nebyl při výměně dostatečně rychlý nebo jsem omylem zasunul vybité monočlánek. Zařízení potom ztratilo údaje pro připojení a bylo nutné ho do sítě Z-Wave opět přidat. Při přidání obdrželo první následující přirozené číslo za nejvyšším použitým Device ID.



Obrázek 4.2. Z-Wave termostat na radiátoru


Varování na nutnost výměny monočlánek se zobrazí ikonkou přímo na termostatech. Případně při úplném vybití monočláneků se přestanou zobrazovat informace na integrovaném displeji a přestanou se aktualizovat informace o tomto zařízení na Z-Wave serveru. Na fotografii 4.2 je ukázka displeje na termostatu rozsvíceného po stisku ovládacího prvku, který má cílovou teplotu nastavenou na 21 °C, je připojen do Z-Wave sítě a má zapnutý „dětský zámek“. Stav nabití monočláneků je také možné kontrolovat průběžně na <http://hub:8083/expert/#/device/battery>. Na obrázcích 4.3 a 4.4 je vidět výřez pro tento termostat v době, kdy měl ještě přidělené Device ID 2. Na prvním obrázku 4.3 je bezpečná hodnota nabití na 55%, ale na druhém obrázku 4.4, je to již kritická hodnota 5% a monočlánek je nutné vyměnit.

#	2
Device Name	radiator 1
Battery type	
Level	 55%
Date	16:46
<input type="button" value="Update"/>	

Obrázek 4.3. Monočlánek u radiátoru č. 2 nabitý na 55%

<sup>1</sup> Je možné vybírat zařízení také přes výrobce, ale to jsem nemusel použít.

<sup>2</sup> Nutné udělat pro zařízení, která již byla v nějaké Z-Wave síti zapojena.

#	2
Device Name	radiator 1
Battery type	
Level	 5%
Date	16:40
<input type="button" value="Update"/>	

**Obrázek 4.4.** Monočlánky u radiátoru č. 2 nabité na 5%

Při instalaci termostatu jsem použil dodávané jednorázové monočlánky - dva kusy AA tužkových baterií. Následně jsem přešel na dobíjitelné monočlánky eneloop Panasonic BK-3MCCE Ni-MH 1900 mAh AA (HR6) a jejich nabíječku Panasonic Ni-MH CHARGER BQ-CC55E.

Na obrázcích 4.3 a 4.4 je také zachyceno `Device Name`. Pro každé zařízení jsem ihned po jeho přidání do Z-Wave sítě využil možnost mu přidělit jméno do `Device Name`. Pro tento první instalovaný termostat jsem po instalaci na <http://hub:8083/expert/#/configuration/interview/2> nastavil jméno dle typu zařízení a jeho pořadí v rámci stejného typu zařízení. Tehdy to bylo na obrázcích 4.3 a 4.4 zobrazené pojmenování *radiator 1*. Později jsem přešel na nové pojmenování, které obsahuje u termostatů také velikost radiátoru a místnost. V březnu 2021 se toto zařízení jmenuje *termostat 1 - velký radiátor v obyvaku*.

Expertní URL na Z-Way serveru s uvedením `Device ID` konkrétního zařízení umožňuje kromě nastavení jména také vynutit dokončení procesu interview. V některých případech<sup>1</sup> se v rámci přidávání nedokočilo interview na 100%. Potom se například na <http://hub:8083/expert/#/configuration/interview/2> zobrazovalo u `Interview` místo 100% jen 90%. Následně jsem mohl pomocí `View Interview Result` volby dohledat `CommandClass` bez zeleného statusu a pomocí `Force Interview` vyžádat interview pro tuto chybějící část a dokončit tak `Interview` na 100%. Tlačítko `Force Interview` pro celé zařízení se mi neosvědčilo. Pokud neuspělo přidání zařízení hned poprvé na 100%, tak tato volba nepomohla.

Jako další instalované Z-Wave zařízení byla chytrá zásuvka FIBARO Wall Plug FGWPF-102-5. Obdržela `Device ID` 3 a toto číslo jí zůstalo od zapojení do Z-Wave sítě až do března 2021. Chytrá zásuvka je trvale napájena ze sítě, a tak se u ní nikdy nestalo, že by ztratila informace pro připojení do sítě. Na obrázku 4.5 je vyfocena chytrá zásuvka ovládající odvlhčovače ve sklepě.

<sup>1</sup> Nejčastěji, když bylo přidávané zařízení daleko od řídicí jednotky.



**Obrázek 4.5.** Z-Wave zásuvka ve sklepě

Následující a poslední zařízení z první dodávky byl chytrý 230V spínač FIBARO Double Switch 2 FGS-223. Na jeho instalaci jsem musel zajistit elektrikáře, neboť bylo nutno také upravit přívod elektřiny k elektrickému kotli. Spínač umí ovládat dvě zařízení. Elektrikář zapojil první spínač na elektrický kotel ústředního topení. Druhý spínač je volný a vhodným kandidátem na jeho použití je hned vedle umístěný bojler na teplou vodu.

Na obrázku 4.6 je vyfocený malý instalační rozvaděč Spelsberg EK 002 IP 54. Uvnitř v levé části je vidět část chytrého spínače FIBARO Double Switch 2 FGS-223 a napravo je EATON xPOLE Z-R230/S. Pod rozvaděčem je jednopólový spínač. Tento způsob zapojení umožňuje v případě selhání Z-Wave sítě manuální zapnutí či vypnutí elektrického kotle spínačem. Manuální přepnutí spínače jsem doposud nemusel použít.



**Obrázek 4.6.** Z-Wave spínač elektrického kotle

Během provozu jsem narazil na nespolehlivost sítě Z-Wave, kterou nebylo nutno řešit manuálním zásahem, ale ošetřil jsem ji na úrovni software. Docházelo k situacím, kdy Z-Wave síť řádně neaktualizovala stav zařízení a například reportovala o vypnutém elektrickém kotli, že je zapnutý. Implementoval jsem proto do mého řídicího software RaZberry Home kontrolní mechanismus, který umí tuto chybu detekovat a změnit status na správný. Na výpisu níže je vidět zachycení takové situace u spínače topení. Technické detaily implementace řešení jsou uvedeny dále v podkapitole 4.5.

```

WARNING: stored as ON, but OFF
Count of mismatches 4
  Device ID: 13, Name: topení, ovládání elektrického kotle
  Element ID: ZWayVDev_zway_13-1

```

Po zapojení a ověření funkčnosti prvních Z-Wave zařízení jsem dokoupil a zapojil další kusy, abych pokryl celý dům. K březnu 2021 mám v provozu celkem 14 Z-Wave zařízení zapojených do jedné sítě. Na Z-Way serveru je možné přes <http://hub:8083/expert/#/device/type> zobrazit jednotlivé typy zařízení. Přehled typů a jejich počty vychází následovně:

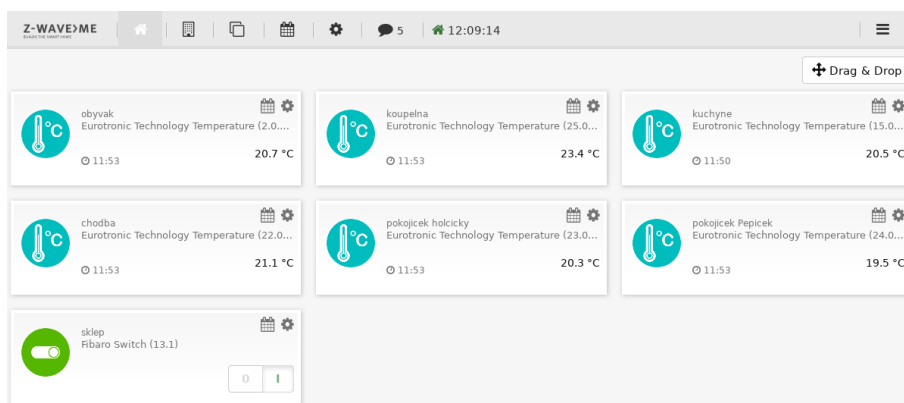
- 1x Static PC Controller - řídicí jednotka
- 6x Binary Power Switch - 5x zásuvka a 1x spínač u elektrického kotle
- 7x General Thermostat V2 - termostaty u radiátorů

Zásuvek jsem po domě zapojil postupně více, než kolik jich využívám na ovládání odvlhčovačů, neboť síť Z-Wave při jedné řídicí jednotce, jedné zásuvce ve středu domu v chodbě, jedné zásuvce ve sklepě, spínači elektrického kotle ve sklepě a chytrých termostatech na radiátorech po obvodu domu vykazovala chyby. Po doplnění domu dalšími zásuvkami (viz náskres domu 4.1) se začala síť chovat spolehlivě.

Další problém, na který jsem narazil, byla instalace termostatu do radiátoru v jídelně. Chytré termostaty po připojení do Z-Wave sítě následně realizují mechanický test, při kterém radiátor zcela otevřou a následně zcela uzavřou. Tento test u radiátoru v jídelně opakovaně neuspěl. Zkusil jsem různé kusy termostatů a měl jsem doma dvakrát instalatéra. Ten dokonce při druhé návštěvě vyměnil část radiátoru, do které se termostat zapojuje. Bohužel ani to nepomohlo a poslední možností zůstala výměna celého radiátoru, což jsem nerealizoval. Chytrý termostat v jídelně tedy zůstal zapojený pouze pro účely jednoho z prvků Z-Wave sítě, ale není možné ho používat jako ostatní termostaty. Radiátor v jídelně není omezený ve vytápění. Vzhledem k tomu, že je jídelna nejchladnější místnost horního patra, tak to nepůsobí komplikace a radiátor jsem zatím nevyměnil.

Co se týče otevřených termostatů, dle instalatéra musí být v domě alespoň jeden radiátor, jehož teplota není termostatem omezena. Jako druhý takový jsem vybral radiátor v chodbě, na kterém je nastavena maximální možná teplota 27,5 °C.

Stav zařízení chytré domácnosti je možné sledovat na webovém rozhraní Z-Way serveru. Na obrázku 4.7 je zobrazena aktuální teplota v obýváku, koupelně, kuchyni, dětských pokojích a v chodbě. Dále je vidět, že spínač elektrického kotle ve sklepě je sepnutý. Elektrický kotel je možné kliknutím na přepínač 0/1 vypnout/zapnout.



Obrázek 4.7. Dashboard na webovém rozhraní Z-Way serveru

Stejný dashboard poskytuje výrobce RaZberry také v cloudu na <https://find.z-wave.me>. V závěru roku 2017 jsme na delší dobu opustili dům. Nebylo ekonomické dům během naší nepřítomnosti vytápět, ale bylo potřeba zajistit, aby v případě silných mrazů nedošlo k zamrznutí vody a poškození domu. Měl jsem v plánu využít tento z Internetu dostupný dashboard na kontrolu stavu a v případě déle trvajících mrazů přes něj zapnout na dálku vytápění. Bohužel dashboard přestal fungovat a já ztratil kontrolu na domácnosti. Naštěstí dům tuto situaci přečkal bez poškození. Pro mne to byla užitečná zkušenost, která mne vedla k tomu, že nesmím být na cloudu závislý a později při plánování této práce jsem *nezávislost na cloudu* také navrhl do podcíle mojí diplomové práce.

Po návratu domů jsem kontaktoval technickou podporu s dotazem na dostupnost API či knihoven pro možnost ovládání RaZberry automatizovanou formou. Podpora výrobce RaZberry mne odkázala na Z-Way User and Programmer Manual [5] a potvrdila problémy s dostupností jejich cloudových služeb.

```
Date: Thu, 4 Jan 2018 11:37:36 +0100
From: Razberry <razberry@z-wave.me>
Subject: Re: API / libraries for z-way-server on RAZBERRY?
To: Jiri Navratil <jiri@navratil.cz>

Am 03.01.18 um 13:34 schrieb Jiri Navratil:
> You mean find as http://find.z-wave.me ? That's service is giving me
> 95% time error.

Unfortunatly true. the server is overleaded and we work at the moment to
split the service to get it back reliable

> Anyway, I can do that from local network. Can the authentication be
> added to the URL? Is API somewhere described?

the only APIDescription is razberry.zwaveme/essentials and of course the
sources in github

> Thank you,
> Jiri

Z-Wave Europe GmbH - Support
fon: +49 3723 8099068
fax: +49 3723 80990 53
Z-Wave Europe GmbH - Antonstr. 3 - 09337 Hohenstein-Er.
hrb chemnitz 18307
md: Elke Geisler
```

## 4.4 Prvotní automatizace

První rychlou možností, jak z Internetu kontrolovat a řídit chytrou domácnost bez závislosti na cloudových službách, by byla možnost využití již dostupného `ssh` přístupu a použít skrze něho buď přesměrování X11 nebo přesměrování portů. Obojí (obzvláště X11 provoz), by však vyžadovalo značnou datovou zátěž.

Rozhodl jsem se pro variantu méně náročnou na datový tok. Místo použití webového rozhraní Z-Way serveru jsem začal hledat způsob, jak zavolat pouze API Z-Way serveru. Díky dokumentaci [5] se mi podařilo tento způsob nalézt a na `rpi04.navratil.lan`

jsem následně vytvořil několik skriptů, pomocí kterých jsem byl schopen API využívat. Nově jsem začal kontrolovat a ovládat domácnost z příkazového řádku. Níže je výpis těchto skriptů a ukázka jejich volání. Název každého skriptu odpovídá jeho určení. Některé skripty ovládají více zařízení.

```
cat /home/navratil/bin/info_topeni.sh
#!/bin/sh
echo Je topeni zapnuto?
curl -u admin:heslo http://hub:8083/ZWaveAPI/Run/\
devices%5B13%5D.instances%5B1%5D.commandClasses%5B37%5D.data.level.value
echo

/home/navratil/bin/info_topeni.sh
Je topeni zapnuto?
true
```

Při zavolání příkazu `curl` s přepínačem `-v` je možné vidět detaily komunikace.

```
curl -v -u admin:heslo http://hub:8083/ZWaveAPI/Run/\
devices%5B13%5D.instances%5B1%5D.commandClasses%5B37%5D.data.level.value
*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to hub (127.0.0.1) port 8083 (#0)
* Server auth using Basic with user 'admin'
> GET /ZWaveAPI/Run/devices%5B13%5D.instances%5B1%5D.
commandClasses%5B37%5D.data.level.value HTTP/1.1
> Host: hub:8083
> Authorization: Basic YWRtaW46c2VXMmZFa1Vw
> User-Agent: curl/7.52.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: Mongoose/6.4
< Content-Type: application/json
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
< Access-Control-Allow-Headers: Authorization
< Connection: keep-alive
< Content-Length: 5
<
* Curl_http_done: called premature == 0
* Connection #0 to host hub left intact
true
```

```
cat /home/navratil/bin/zapni_topeni.sh
#!/bin/sh
curl -u admin:heslo http://hub:8083/ZAutomation/api/v1/devices/\
ZWayVDev_zway_13-1-37/command/on # zapni topeni
echo

/home/navratil/bin/zapni_topeni.sh
{"data":null,"code":200,"message":"200 OK","error":null}
```

```
cat /home/navratil/bin/vypni_topeni.sh
#!/bin/sh
```

```
curl -u admin:heslo http://hub:8083/ZAutomation/api/v1/devices/\
ZWayVDev_zway_13-1-37/command/off # vypni topeni
echo
```

```
/home/navratil/bin/vypni_topeni.sh
{"data":null,"code":200,"message":"200 OK","error":null}
```

```
cat /home/navratil/bin/zapni_odvlhcovace.sh
#!/bin/sh
curl -u admin:heslo http://hub:8083/ZAutomation/api/v1/devices/
ZWayVDev_zway_20-0-37/command/on # pracovna
echo
curl -u admin:heslo http://hub:8083/ZAutomation/api/v1/devices/
ZWayVDev_zway_21-0-37/command/on # koupelna
echo
curl -u admin:heslo http://hub:8083/ZAutomation/api/v1/devices/
ZWayVDev_zway_30-0-37/command/on # sklep
echo
```

```
/home/navratil/bin/zapni_odvlhcovace.sh
{"data":null,"code":200,"message":"200 OK","error":null}
{"data":null,"code":200,"message":"200 OK","error":null}
{"data":null,"code":200,"message":"200 OK","error":null}
```

```
cat /home/navratil/bin/vypni_odvlhcovace.sh
#!/bin/sh
curl -u admin:heslo http://hub:8083/ZAutomation/api/v1/devices/
ZWayVDev_zway_20-0-37/command/off # pracovna
echo
curl -u admin:heslo http://hub:8083/ZAutomation/api/v1/devices/
ZWayVDev_zway_21-0-37/command/off # koupelna
echo
curl -u admin:heslo http://hub:8083/ZAutomation/api/v1/devices/
ZWayVDev_zway_30-0-37/command/off # sklep
echo
```

```
/home/navratil/bin/vypni_odvlhcovace.sh
{"data":null,"code":200,"message":"200 OK","error":null}
{"data":null,"code":200,"message":"200 OK","error":null}
{"data":null,"code":200,"message":"200 OK","error":null}
```

Skripty výše jsem mohl volat z příkazové řádky. V rámci první jednoduché formy automatizace jsem je také použil na nastavení režimu vytápění a ovládání odvlhčovačů v rámci týdne. Nastavení crontab pro volání skriptů v rámci služby cron vypadalo následovně.

```
crontab -l
MAILTO=jiri@navratil.cz
0 6 * * * /home/navratil/bin/zapni_topeni.sh
30 7 * * * 1-5 /home/navratil/bin/vypni_topeni.sh
0 11 * * * 1-5 /home/navratil/bin/zapni_topeni.sh
0 22 * * * /home/navratil/bin/vypni_topeni.sh
30 10 * * * /home/navratil/bin/zapni_odvlhcovace.sh
30 15 * * * /home/navratil/bin/vypni_odvlhcovace.sh
```

Na základě tohoto nastavení pro topnou sezónu se každý den v 6:00 zaplo topení. V pracovní dny bylo topení vypnuté od 7:30 do 11:00 a každý den se vypnulo ve 22:00. Odvlhčovače se zapínaly denně od 10:30 do 15:30.

## 4.5 RaZberry Home

Tato podkapitola popisuje, jak jsem přešel od první jednoduché automatizace zprovozněné v minulé podkapitole k plně automatizovanému řešení. Toto řešení jsem nazval RaZberry Home a používám pro něj zkratku `rh`.

Už automatizace zavedaná v předešlé podkapitole mi umožnila nezávislost na cloudové službě výrobce RaZberry. V nastavení Z-Way serveru na `http://hub:8083/smarthome/#/admin` jsem vypnul `Remote Access`. Výrobce mne od toho okamžiku začal při každém přístupu do nastavení varovat, že nemám vzdálený přístup nastavený a nabádá mne, abych ho aktivoval: „Remote Access is not activated. Please activate it in Apps-Active tab.“

Dále jsem na Raspberry Pi vypnul Wi-Fi rozhraní. Součástí instalace Z-Way serveru bylo také aktivování AP na Wi-Fi rozhraní. Toto rozhraní jsem nijak nevyužíval, a tak bylo bezpečnější ho vypnout.

### 4.5.1 InfluxDB

Pro vylepšenou formu automatizace jsem předpokládal častou práci s daty ze zařízení. V předešlém kroku jsem se naučil číst stav zařízení a zařízení ovládat. U další úrovně automatizace předpokládám časté dotazy do sítě a proto místo častého čtení přímo ze zařízení budu ukládat stav průběžně do databáze a dotazy budou směřovány do databáze nikoli na zařízení.

Z-Way server umožňuje instalaci doplňkových modulů. Našel jsem modul `InfluxDb Statistics Collector` pro ukládání dat ze zařízení do `InfluxDB`. Rozhodl jsem se tedy na `rpi04.navratil.lan` nainstalovat `InfluxDB`.

```
sudo apt install apt-transport-https
curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add -
echo "deb https://repos.influxdata.com/debian jessie stable" | \
    sudo tee /etc/apt/sources.list.d/influxdb.list
sudo apt update
sudo apt install influxdb
```

Po instalaci jsem vypnul odesílání reportů na `usage.influxdata.com`, zapnul jsem autentizaci, změnil jsem defaultní politiku promazávání a `influxdb` jsem spustil.

```
sudo vi /etc/influxdb/influxdb.conf
reporting-disabled = true
auth-enabled = true
retention-autocreate = false
sudo service influxdb start
```

Následovalo vytvoření datáze a uživatelů.

```
influx
Connected to http://localhost:8086 version 1.4.2
InfluxDB shell version: 1.4.2
> CREATE DATABASE zwave
> SHOW DATABASES
name: databases
```



```

name
----
_internal
zwave
> CREATE RETENTION POLICY trvale ON zwave DURATION INF
  REPLICATION 2 DEFAULT
> SHOW RETENTION POLICIES ON zwave
name    duration shardGroupDuration replicaN default
-----
trvale 0s          168h0m0s          2          true
> CREATE USER navratil WITH PASSWORD 'heslo' WITH ALL PRIVILEGES
> CREATE USER script01 WITH PASSWORD 'heslo'
> GRANT WRITE ON zwave TO script01
> SHOW USERS
user     admin
-----
navratil true
script01 false

```

Zajištění spuštění po rebootu.

```

root@rpi04:~# vim /etc/rc.local
sleep 10
echo Starting InfluxDB
/usr/bin/influxd -config /etc/influxdb/influxdb.conf

```

Na Z-Way serveru jsem přes `http://hub:8083/smarthome/#/apps/instance` aktivoval a nastavil modul pro ukládání dat do InfluxDB.

- Name: InfluxDb Statistics Collector
- Database Server URL: `http://127.0.0.1`
- Database Server Port: 8086
- Database name: zwave
- Database username: script01
- Database password: heslo
- Tags: influxdb
- Update Interval : 5 minutes

U všech zařízení, jejichž hodnoty jsem chtěl ukládat do InfluxDB, jsem přidal tag `influxdb`. Například pro teplotu na termostatu v obyvaku s Device ID 35 jsem přidal tag `influxdb` na `http://hub:8083/smarthome/#/element/ZWayVDev_zway_35-0-49-1`.

Lokálně je možné na počítaři `rpi04.navratil.lan` spustit InfluxDB shell pomocí příkazu `influx`, zvolit databázi `zwave` a pomocí dotazů sledovat aktuální (samozřejmě je možné zjišťovat i historický) stav zařízení, jako například elektrického topení ve sklepe (level 0 znamená vypnutý) a nebo teplotu na termostatu v obyvaku.

```

influx
Connected to http://localhost:8086 version 1.4.2
InfluxDB shell version: 1.4.2
> USE zwave
Using database zwave
> SELECT time,level,room FROM "device.ZWayVDev_zway_13-1-37"
  ORDER BY DESC LIMIT 1;
name: device.ZWayVDev_zway_13-1-37

```

```

time                level room
-----
1616958520534832992 0      sklep
> SELECT time,level,room FROM "device.ZWayVDev_zway_35-0-49-1"
ORDER BY DESC LIMIT 1;
name: device.ZWayVDev_zway_35-0-49-1
time                level room
-----
1616958520534832992 20.43 obyvak

```

Data se ukládají do adresáře `/var/lib/influxdb` a log je možné sledovat v souboru `/var/log/syslog`.

Tím je spuštěno ukládání potřebných dat z vybraných zařízení do InfluxDB. Z-Way server ukládá údaje označené tagem `influxdb` každých 5 minut. Moje aplikace si tyto data bude z databáze načítat.

## 4.5.2 Git

Před započítím kódování jsem na mém Git serveru vytvořil repozitář `rh.git`.

```

ssh git@localhost
pwd
/home/git
mkdir rh.git
cd rh.git/
git init --bare
Initialized empty Git repository in /home/git/rh.git/

```

Naklonoval jsem repozitář na notebooku s Git serverem.

```

cd ~/src
git clone git@localhost:rh

```

Následně jsem naklonoval repozitář na Raspberry Pi. Nejprve na `rpi04` a později také na `rpi19`.

```

cd ~/src
git clone git@192.168.4.50:rh

```

Na Raspberry Pi jsem doposud Git nepoužíval. Doplnil jsem tedy jeho konfiguraci o `user.name` a `user.email`.

```

git config --global user.name "Jiri Navratil"
git config --global user.email "jiri@navratil.cz"

```

V rámci tvorby práce jsem pracoval na všech počítačích v adresáři `~/src/rh`. Pravidelně jsem kód ukládal, využíval `git diff` a u commitů jsem se snažil zachytit podstatnou zprávu, která účel commitu co nejlépe vystihla. Do verze 1.0 jsem realizoval 140 commitů.

Nejčastější posloupnost příkazů na stroji s prováděnými změnami.

```

git diff
git add
git status
git commit
git push

```

Následně jsem na jiném stroji postupoval takto, abych se dostal do stavu „Your branch is up to date with 'origin/master“.

```
git fetch
git rebase -i
```

Takto vypadají zprávy z `git log` při finalizaci verze 1.0. Jedná se verzi odevzdanou jako příloha diplomové práce.

```
commit c766c40dc1c8d245ce3737bda839899c4dcb0e98
(HEAD -> master, origin/master, origin/HEAD)
Author: Jiri Navratil <jiri@navratil.cz>
Date: Sat Dec 12 21:18:11 2020 +0100

    12.12.2020 - set version to 1.0

commit ba21ad63da2756f5a8efa394c02934e5f8a2e761
Author: Jiri Navratil <jiri@navratil.cz>
Date: Sat Dec 12 16:41:03 2020 +0100

    MkDocs initial files

git log | grep commit | wc --lines
140
```

### 4.5.3 Verze 1.0

Práce na první verzi RaZberry Home (zkráceně `rh`) jsem začal přímo na stroji s RaZberry označeným `rpi04.navratil.lan` a aliasem `hub`. Později jsem s požadavkem na rozšíření řešení o LCD displej a tudíž a nasazení dalšího počítače Raspberry Pi přesunul běh `rh` na Raspberry Pi s LCD volným GPIO pro LCD Displej označené jako `rpi19.navratil.lan`.

Pro potřeby převedení mých poznámek do textu práce budu rovnou vycházet ze stavu kódování a běhu `rh` na `rpi19`. Toto zjednodušení by nemělo vliv na reprodukovatelnost. Při opakování řešení zcela od začátku bych se znalostí potřeby LCD dsipleje rovnou rozvhl práci takto.

Bylo příjemné a čas šetřecí to, že návrh `rh` umožňoval jednoduchý přechod z původního localhost na vzdálený přístup.

Po prvním naklonování repozitáře jsem do Raspbian OS doinstaloval potřebné vývojářské balíčky, zprovoznil virtuální prostředí `venv` v Python 3 a nainstaloval potřebné Python moduly.

```
apt install libpq-dev python3-dev
python3 -m venv .venv --prompt=rh
. .venv/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
```

Tady je přehled modulů specifikovaných v textovém souboru `requirements.txt`.

```
cat requirements.txt
influxdb
psycopg2-binary
pycodestyle
mkdocs
```

### ■ 4.5.4 Modul rh (main)

Chtěl jsem napsat aplikaci, která poběží v nekonečné smyčce, bude kontrolovat a řídit domácnost a o stavu bude v krátkých intervalech informovat na terminálu. Část z těchto požadavků řeší v OpenBSD program iostat [40], který monitoruje stav disků a ve zvolených intervalech informace vypočítává na terminál a zvládá i zajistit, aby byly informace nadepsány na začátku a dále vždy po 20 vypsaných řádcích s detaily o discích.

Inspiroval jsem se zdrojovým kódem [40] v jazyce C a napsal tuto smyčku v jazyce Python. Níže je kód z `rh.py` za řádkem `if __name__ == '__main__':`. Kvůli šířce papíru je odsazení ve výpisu níže sníženo o čtyři mezery. Dále jsem upravil zalamování některých řádků. Původní používané formátování je vidět ve zdrojovém kódu v odevzdané příloze.

```

today = datetime.date.today()

wanthead, hrcnt = False, 1
while True:
    hrcnt -= 1
    inverter.read_data()
    if (not hrcnt or wanthead):
        # vytiskni hlavičku (nadpis pro hodnoty)
        print("{} {} {}".format(
            inverter.header(),
            device_module.Device.header(),
            ":%d.%m.%Y".format(datetime.date.today())
        ), flush=True)
        elektricky_kotel.double_check()
        wanthead, hrcnt = False,
            cfg_number_of_lines_before_next_header
    # vytiskni informační řádek s hodnotami
    print("{} {} {:>13}".format(
        inverter.info(),
        device_module.Device.info(radiator_v_obyvaku, radiator_v_kuchyni,
            radiator_u_kluku, radiator_u_holek, radiator_v_koupelne,
            radiator_v_chodbe, elektricky_kotel
        ),
        ":%H:%M:%S".format(datetime.datetime.now())
    ), flush=True)
    # akce (pokyny) pro zařízení
    if inverter.consume_energy() or cfg_heating.is_heating_season:
        # odvlhčovače zapni vždy v topné sezóně
        # a mimo ni jen když je dostatek vyráběné energie
        odvlhcovace.on()
    elif inverter.save_energy() and not cfg_heating.is_heating_season:
        # mimo topnou sezónu vypni odvlhčovače,
        # pokud je potřeba šetřit energii
        odvlhcovace.off()
    if not radiatory.zapnout_kvuli_mrazu():
        # v topne sezone, kdyz nehrozi poskozeni mrazem
        # vypni nebo zapni elektrokotel
        radiatory_pro_rizeni_kotle.zapnout_nebo_vypnout_kotel()
    if today != datetime.date.today():
        # kvůli ukládání extrémů zjišťuji, zda je nový den
        today = datetime.date.today()

```

```
# resetuj denní maxima
inverter.reset_extrema()

time.sleep(1)
```

Kód výše v sobě zachycuje základní algoritmus. V textu dále ukáži některé detaily a vypíši klíčové či zajímavé části kódu. Opět uvádím, že kompletní kód je součástí přílohy.

Ukázka z běhu kódu (levá část)

```
88% nabíjení 96% 100% 600W -384W 17W nakupuji -172W 60W
88% nabíjení 98% 100% 600W -376W 9W nakupuji -172W 60W
nabití autonomie má sp. výroba spotřeba síť baterie součet
88% nabíjení 98% 100% 600W -376W 9W nakupuji -172W 60W
88% nabíjení 100% 97% 597W -354W -11W prodávám -176W 56W
88% nabíjení 100% 99% 597W -371W -4W prodávám -166W 56W
88% nabíjení 100% 94% 597W -354W -21W prodávám -166W 56W
88% nabíjení 100% 100% 601W -378W 1W nakupuji -163W 61W
```

a odpovídající pravá část.

```
19.0 C 18.2 C 15.4 C 15.4 C 15.7 C 16.3 C VYP 16:30:01
19.0 C 18.2 C 15.4 C 15.4 C 15.7 C 16.3 C VYP 16:30:03
obyvák kuchyň kluci holky koupelna chodba topení 13.05.2020
19.0 C 18.2 C 15.4 C 15.4 C 15.7 C 16.3 C VYP 16:30:04
19.0 C 18.2 C 15.4 C 15.4 C 15.7 C 16.3 C VYP 16:30:07
19.0 C 18.2 C 15.4 C 15.4 C 15.7 C 16.3 C VYP 16:30:08
19.0 C 18.2 C 15.4 C 15.4 C 15.7 C 16.3 C VYP 16:30:11
19.0 C 18.2 C 15.4 C 15.4 C 15.7 C 16.3 C VYP 16:30:12
```

Hlavička je v ukázce záměrně na třetím řádku. Při běhu aplikace se hlavička vypíše na spodním řádku a následně se postupně odsouvá a ve správný čas (přesněji řečeno při proměnné `hdrcnt` rovné nule) se znovu vypíše. Počet řádků `cwmezi` hlavičkami je nastavený v `cfg_number_of_lines_before_next_header`. Proměnnou `wanthead` k vyžádání hlavičky jsem zatím nepotřeboval.

Ukázka výstupu aplikace pochází ze 13. května 2020. Tedy mimo topnou sezónu. Sloupečky pro oba výstupy zleva doprava.

- nabití - stav nabití baterií z intervalu 7%<sup>1</sup> a 100%
- nabíjení / vybití - jednoslovný popis toho, zda se baterie nabíjí nebo vybíjí
- autonomie - nezávislost na elektřině ze sítě, střídač vyrovnává změny ze sítě, proto je na prvním řádku autonomie 96%, přestože je výroba 600W vyšší než spotřeba -384W
- má sp. - má spotřeba - 100% znamená, že veškerou elektřinu spotřebovávám přímou spotřebou a nabíjením akumulátorů
- výroba - aktuální výroba solární energie
- spotřeba - aktuální spotřeba spotřebiči v domě
- síť - kladné číslo znamená nákup ze sítě, záporné číslo znamená prodej do sítě
- nakupuji / prodávám - slovní popis jako přehledné doplnění kladné versus záporné číslo z předchozího sloupce
- baterie - záporné nabíjení, kladné vybíjení (čerpání z akumulátorů)
- součet - ztráta energie při přesunech
- obyvák - teplota v obyvaku
- kuchyň - teplota v kuchyni

<sup>1</sup> Střídač nepovolí vybíjení baterií pod 7%.

- kluci - teplota v pokoji Pepíčka
- holky - teplota v pokoji u holčiček
- koupelna - teplota v koupelně
- chodba - teplota v chodbě
- topení - VYP = vypnuté, ZAP = zapnuté
- datum - hlavička obsahuje datum
- čas - informační řásky obsahují čas

### ■ 4.5.5 Modul config

Konfigurace je uložena v souboru config.py. Pro čtení atributů využívám @property.

```
class ConfigInverter:
    u""" Třída s konfigurací pro Fronius střídač """
    def __init__(self):
        self._ip = "192.168.4.8"
        self._timeout = 3

    @property
    def host(self):
        return self._ip

    @property
    def timeout(self):
        return self._timeout
```

V rh.py potom použiji

```
import config
import inverter as inverter_module

cfg_inverter = config.ConfigInverter()
inverter = inverter_module.Inverter(
    cfg_inverter.host, cfg_inverter.timeout, rdb
)
```

Pro debugování kódu je možné nastavit proměnnou prostředí DEBUG. Do kódu je potom přebírána tímto způsobem `DEBUG = os.environ.get('DEBUG', False)`

Pro vysvětlení, jaké parametry rozhodují o tom, zda se bude vytápět je důležitá třída ConfigHeating.

```
class ConfigHeating:
    u""" Třída s konfigurací pro vytápění """
    def __init__(self):
        self._heating_season = True
        self._maximum_of_mismatches = 10
        # lowest safe temperature during heating season
        # at least 5 degrees C
        self._frost_protection = 5
        # bude dům vytápěn pro ženu?
        # pro ženy jsou teploty pro vytápění posunuty o 1 stupeň C
        self._thermal_comfort_for_women = True
        if self._thermal_comfort_for_women:
            self._heating_max_temperature = 21
            self._heating_min_temperature = 20
```

```

else:
    self._heating_max_temperature = 20
    self._heating_min_temperature = 19
# interval pro vytápění o víkendu
self._heating_during_weekend = (
    (datetime.time(4, 00), datetime.time(21, 00)),
)
# intervaly pro vytápění v pracovní dny
self._heating_during_working_day = (
    (datetime.time(4, 00), datetime.time(7, 30)),
    (datetime.time(12, 00), datetime.time(20, 30)),
)
# vytápění pro státní svátek v pracovní den
# nebo pro práci z domova
# tuple je doplňováno a zkracováno
self._heating_in_these_weekdays = (
    datetime.date(2020, 12, 7),
    datetime.date(2020, 12, 8),
    datetime.date(2020, 12, 9),
    datetime.date(2020, 12, 10),
    datetime.date(2020, 12, 11),
)
# nikdo doma, nevytápět
self._no_heating_in_these_days = (
    datetime.date(2020, 12, 1),
    datetime.date(2020, 12, 2),
    datetime.date(2020, 12, 3),
    datetime.date(2020, 12, 4),
    datetime.date(2020, 12, 5),
)

@property
def maximum_of_mismatches(self):
    return self._maximum_of_mismatches

@property
def is_heating_season(self):
    return self._heating_season

@property
def frost_protection(self):
    return self._frost_protection

@property
def heating_min_temperature(self):
    return self._heating_min_temperature

@property
def heating_max_temperature(self):
    return self._heating_max_temperature

@property
def intervals_for_heating(self):

```

```

_now = datetime.datetime.now()
# is today an exception? (no heating)
if _now.date() in self._no_heating_in_these_days:
    if DEBUG:
        print("no heating today (no intervals for heating)")
    return tuple()
# is today other exception? (public holiday in a weekday)
if _now.date() in self._heating_in_these_weekdays:
    if DEBUG:
        print(
            "intervals for heating",
            self._heating_during_weekend
        )
    return self._heating_during_weekend

# is now weekend?
# public holidays are not covered
if _now.weekday() > 4:
    # weekend
    return self._heating_during_weekend
else:
    # working day
    return self._heating_during_working_day

```

Vytápění se zapíná pro topnou sezónu pomocí proměnné `_heating_season`. V rámci týdne je jiný režim pro pracovní den a jiný pro víkend. V pracovní den se netopí v době, kdy jsou děti ve škole a dospělí v práci. Pro státní svátek nebo práci z domova je možné tento režim upravit na „celodenní“ přidáním datumu (například `datetime.date(2020, 12, 8)` pro 8. prosinec 2020) do tuple `_heating_in_these_weekdays`. Naopak pro situaci, že doma celý den nikdo nebude, je možné využít tuple `_no_heating_in_these_days`. Při přítomnosti ženy v domě je možné zvednout tepelný komfort například o 1 °C. Ve dnech s vypnutým vytápěním se kontroluje, aby teplota neklesla pod `_forst_protection`.

Během provozu jsem zjistil, že se někdy správně neaktualizuje status vypnutý / zapnutý elektrického kotle. Zavedl jsem proto kontrolní mechanismus. Proměnná `_maximum_of_mismatches` definuje, kolik je potřeba nalezených chyb pro vynucenou změnu stavu.

Intervaly pro vytápění `intervals_for_heating(self)` se využívají ve funkci `is_time_for_heating()` jestli je správný čas pro vytápění.

```

def is_time_for_heating():
    _time = datetime.datetime.now().time()
    for i in cfg_heating.intervals_for_heating:
        if i[0] < _time < i[1]:
            return True
    return False

```

## 4.5.6 Modul device

Tento modul definuje třídu `Device`, metody a atributy pro použití se Z-Wave zařízeními. Třída `Device` mimo jiné definuje množinu `all_instances` a v rámci `__init__` do ní přidává nové instance voláním `Device.all_instances.add(self)`. Na úrovni třídy jsou definovány tři metody `@classmethod`



- `get_set_of_instances(cls)` - vrací množinu všech instancí
- `header(cls)` - hlavička pro tisk na terminál
- `info(cls, radiator_v_obyvaku, ..)` - informační řádek pro tisk na terminál

Pro použití s objekty jsou definovány tyto třídy

- `update_temperature_from_influxdb(self)` - načítá teplotu z InfluxDB
- `get_temperature(self)` - vrací teplotu jako reálné číslo
- `get_temperature_as_string(self)` - vrací teplotu jako řetězec včetně formátování se °C.
- `is_on(self)` - vrací True pro zapnuto a False pro vypnuto
- `is_on_or_off_as_string(self)` - vrací ZAP pro zapnuto a VYP pro vypnuto
- `check_status_via_url(self)` - kontrola stavu přímo ze zařízení kvůli detekování neaktualizovaného stavu
- `execute_url_command(self, url)` - spuštění příkazu zavoláním url
- `on(self)` - zavolání url pro zapnutí
- `off(self)` - zavolání url pro vypnutí
- `double_check(self)` - kontrola uloženého stavu a vyžádané stavu ze sítě
- `print(self)` - tisk pro debugování

### 4.5.7 Modul room

Tento modul definuje třídu Room pro práci s pokoji. Instance třídy Device se přidávají do pokojů pro určení lokality. Oproti předešlým třídám jsou tu nově použity metody třídy @classmethod pro vrácení prvků množiny všech instancí seříděné podle atributu `_device_id` případně `_name`.

```
@classmethod
def get_list_of_instances_sorted_via_id(cls):
    return sorted(cls.all_instances,
                  key=operator.attrgetter('_device_id'))

@classmethod
def get_list_of_instances_sorted_via_name(cls):
    return sorted(cls.all_instances,
                  key=operator.attrgetter('_name'))
```

### 4.5.8 Modul hub

Třída pro práci se Z-Way serverem běžícím nad RaZberry na Raspberry Pi počítači `rpi04.navratil.lan` s alisem hub.

Třída Hub obsahuje mimo jiné tři atributy, které jsou zapsané ve formě šablony

```
self._url_info_level =
"http://{ip}:{port}/ZWaveAPI/Run/devices%5B{device_id}%5D.instances%5B{device_id}%5D"
".commandClasses%5B{device_id}%5D.data.level.value".format(
    self._ip, self._port, "{ip}", "{port}", "{device_id}")
)
self._url_command_on =
"http://{ip}:{port}/ZAutomation/api/v1/devices/ZWayVDev_zway_{device_id}-{device_id}"
"/command/on".format(
    self._ip, self._port, "{ip}", "{port}", "{device_id}")
)
self._url_command_off =
```

```
"http://{ip}:{port}/ZAutomation/api/v1/devices/ZWayVDev_zway_{id}-{id}-{id}"
"/command/off".format(
    self._ip, self._port, "{id}", "{id}", "{id}"
)
```

a jsou používány při vytváření nové instance z třídy Device, kde se v `__init__` nastaví konkrétní url a `commandClass` 37

```
self.url_info_level = hub.url_info_level.format(
    self._device_id, self._instance_id, 37
)
self.url_command_on = hub.url_command_on.format(
    self._device_id, self._instance_id, 37
)
self.url_command_off = hub.url_command_off.format(
    self._device_id, self._instance_id, 37
)
```

### ■ 4.5.9 Třída Dehumidifiers

Třída Dehumidifiers se používá pro vytvoření množiny všech odvlhčovačů. Obsahuje metody na zapínání a vypínání odvlhčovačů.

### ■ 4.5.10 Třída Radiatory

Třída Radiatory se používá pro práci s termostaty na radiátorech. Osahuje metody na nepodmíněné zapínání a vypínání radiátorů, metodu na podmíněné zapnutí v případě rizika zamrznutí (viz dříve zmíněný atribut u konfigurace `frost_protection`) a metodu která rozhodne o vypnutí či zapnutí elektrického kotle.

```
def zapnout_kvuli_mrazu(self):
    """ frost protection """
    for i in self.get_set_of_devices():
        if DEBUG:
            print("{id}: {:.1f} C".format(i.Name, i.get_temperature()))
        if i.get_temperature() < cfg_heating.frost_protection:
            elektricky_kotel.on()
            return True
    return False

def zapnout_nebo_vypnout_kotel(self):
    if not cfg_heating.is_heating_season:
        if DEBUG:
            print("Heating season: NO")
        elektricky_kotel.off()
        return
    if not is_time_for_heating():
        if DEBUG:
            print("Time for heating: NO")
        elektricky_kotel.off()
        return
    zapnout_kotel, vypnout_kotel = None, None
    for i in self.get_set_of_devices():
        if DEBUG:
            print("{id}: {:.1f} C".format(i.Name, i.get_temperature()))
```

```

    if i.get_temperature() < cfg_heating.heating_min_temperature:
        zapnout_kotel = i.Description
    elif i.get_temperature() > cfg_heating.heating_max_temperature:
        vypnout_kotel = i.Description
    if zapnout_kotel and vypnout_kotel:
        print(
            "POZOR - požadavek na zapnutí ({} ) i na vypnutí ({} ) kotle"
            " - vypínám kotel".format(zapnout_kotel, vypnout_kotel)
        )
        elektricky_kotel.off()
        return
    if zapnout_kotel:
        elektricky_kotel.on()
    elif vypnout_kotel:
        elektricky_kotel.off()

```

### ■ 4.5.11 Inicializace objektů

V hlavním modulu jsou postupně inicializovány objekty. Následuje část kódu pro vytvoření představy.

```

hub = hub_module.Hub(
    cfg_hub.host, cfg_hub.port,
    cfg_hub.user, cfg_hub.password,
    cfg_hub.timeout
)

```

```

none = room_module.Room(hub, 0, "nepřiřazeno")
sklep = room_module.Room(hub, 1, "sklep")
chodba = room_module.Room(hub, 2, "chodba")
kluci = room_module.Room(hub, 3, "pokojíček kluci")
holky = room_module.Room(hub, 4, "pokojíček holčičky")
koupelna = room_module.Room(hub, 5, "koupelna")
pracovna = room_module.Room(hub, 6, "pracovna")
jidelna = room_module.Room(hub, 7, "jidelna")
kuchyn = room_module.Room(hub, 8, "kuchyň")
obyvak = room_module.Room(hub, 9, "obyvák")

```

```

radiator = Radiatory()
radiator_pro_rizeni_kotle = Radiatory()

radiator_v_obyvaku = device_module.Device(
    hub, 35, 0, "ZWayVDev_zway_35-0", "termostat 1",
    "velký radiátor v obyvaku"
)
obyvak.add_device_to_room(radiator_v_obyvaku)
radiator.add(radiator_v_obyvaku)
radiator_pro_rizeni_kotle.add(radiator_v_obyvaku)

```

```

odvlhcovace = Dehumidifiers()

odvlhcovac_v_pracovne = device_module.Device(
    hub, 20, 0, "ZWayVDev_zway_20-0", "zásuvka 3",
    "zásuvka v pracovně s odvlhčovačem"
)

```

```

)
pracovna.add_device_to_room(odvlhcovac_v_pracovne)
odvlhcovace.add(odvlhcovac_v_pracovne)

odvlhcovac_v_koupelne = device_module.Device(
    hub, 21, 0, "ZWayVDev_zway_21-0", "zásuvka 4",
    "zásuvka v koupelně s odvlhčovačem"
)
koupelna.add_device_to_room(odvlhcovac_v_koupelne)
odvlhcovace.add(odvlhcovac_v_koupelne)

odvlhcovac_ve_sklepe = device_module.Device(
    hub, 30, 0, "ZWayVDev_zway_30-0", "zásuvka 2",
    "zásuvka ve sklepě s odvlhčovačem"
)
sklepe.add_device_to_room(odvlhcovac_ve_sklepe)
odvlhcovace.add(odvlhcovac_ve_sklepe)

```

Obdobně jsou inicializovány i další objekty. Kompletní kód je v příloze.

### ■ 4.5.12 Použití aplikace

Připojení na řídicí počítač je možné pouze z vnitřní sítě. Při potřebě připojení z Internetu je nutné nejprve připojení na veřejně dostupný počítač `astaldo.navratil.cz` a následně na řídicí počítač (původně `rpi04.navratil.lan` a nově `rpi19.navratil.lan` viz v následující kapitole).

```

ssh -A astaldo.navratil.cz # 109.164.115.11
ssh 192.168.4.83 # rpi19.navratil.lan

```

Na řídicím počítači by již měl běžet terminálový multiplexor. Takto se do něho připojuje.

```
tmux attach
```

Pokud by tmux neběžel, tak je potřeba udělat tyto kroky

```

cd ~/src/rh
tmux
. .venv/bin/activate
./rh.py | tee -a /tmp/rh.txt
ctrl+b c
tail -f /tmp/rh.txt | sudo tee /dev/console

```

Nyní by řídicí aplikace `rh` měla běžet a vypisovat aktuální stav na terminál. Přepínání mezi okny je možné pomocí `ctrl+b n`.

Toto jsou pouze stručné instrukce. Detailní návod k použití je součástí MkDocs dokumentace, která je v příloze.

## ■ 4.6 Instalace LCD displeje

Použití terminálového multiplexoru `tmux` je praktické pro mne jako autora práce reprezentující osobu vývojář. Pro osoby děti a žena by se ale hodil nějaký permanentně dostupný displej.

Zakoupil jsem Waveshare 7" LCD displej, 1024x600, IPS, DPI, se stojánkem. Displej je řízený přímo z Raspberry Pi přes RGB LCD HAT modul s DPI rozhraním.

Potřeboval jsem volné GPIO, které již bylo na `rpi04.navratil.lan` obsazené RaZberry modulem. Použil jsem další Raspberry Pi `rpi19.navratil.lan` a LCD displej na něj přes GPIO připojil.

Následuje detailní popis instalace operačního systému, jeho konfigurace a úpravy na routeru.

Pro instalaci operačního systému byl použit Raspbian Buster Lite image a byl umístěn na SanDisk Extreme 64 GB 3 MicroSD XC I kartu. Image byl nakopírován na SD kart v notebooku s OpenBSD.

```
SHA256 (2019-09-26-raspbian-buster-lite.zip) =\
a50237c2f718bd8d806b96df5b9d2174ce8b789eda1f03434ed2213bbca6c6ff

unzip 2019-09-26-raspbian-buster-lite.zip
Archive: 2019-09-26-raspbian-buster-lite.zip
  inflating: 2019-09-26-raspbian-buster-lite.img

sha256 2019-09-26-raspbian-buster-lite.img
SHA256 (2019-09-26-raspbian-buster-lite.img) =\
fef882aa980cda898421e0f71328033c02cc314ed438b5ec8f39b03e641f31b1

umass2 at uhub1 port 3 configuration 1 interface 0 "Generic USB2.0-CRW"
umass2: using SCSI over Bulk-Only
scsibus6 at umass2: 2 targets, initiator 0
sd6 at scsibus6 targ 1 lun 0: <Generic-, SD/MMC/MS PRO, 1.00> SCSI2
sd6: 60906MB, 512 bytes/sector, 124735488 sectors

doas dd if=2019-09-26-raspbian-buster-lite.img of=/dev/rsd6c bs=32k

68608+0 records in
68608+0 records out
2248146944 bytes transferred in 217.461 secs (10338130 bytes/sec)

cmp 2019-09-26-raspbian-buster-lite.img /dev/rsd6c

doas disklabel sd6
# /dev/rsd6c:
type: SCSI
disk: SCSI disk
label: SD64GB
duid: c8949cc4f8525188
flags:
bytes/sector: 512
sectors/track: 63
tracks/cylinder: 255
sectors/cylinder: 16065
cylinders: 7764
total sectors: 124735488
boundstart: 0
boundend: 124735488
drivedata: 0

16 partitions:
#          size          offset  fstype  [fsize bsize  cpg]
```

```

c:          124735488          0  unused
i:          524288            8192  MSDOS
j:          124203008         532480  ext2fs

```

Povolení služby OpenSSH.

```

doas mount_msdos -l -u 5006 /dev/sd6i /mnt/flash
touch /mnt/flash/ssh
doas umount /mnt/flash/

```

Upravení nastavení na routeru s OpenBSD.

```

vi /etc/dhcpd.conf
    host rpi19w {
        hardware ethernet b8:27:eb:d5:f9:ef;
        fixed-address 192.168.4.83;
    }
    host rpi19e {
        hardware ethernet b8:27:eb:80:ac:ba;
        fixed-address 192.168.4.84;
    }
\nedtt

\begtt
vi /etc/hosts
192.168.4.84    rpi19e rpi19
192.168.4.83    rpi19w

vi /etc/ethers
b8:27:eb:80:ac:ba    rpi19e
b8:27:eb:d5:f9:ef    rpi19w

rcctl restart dhcpd rarpd
dhcpd(ok)
dhcpd(ok)
rarpd(ok)
rarpd(ok)

```

Zasunutí SD karty do Raspberry Pi, nabootování a instalace. Původní heslo pro uživatele pi bylo raspberry. Heslo jsem ihned po prvním přihlášení změnil a později jsem uživatele pi smazal.

```

ssh pi192.168.4.83
passwd
sudo apt install tmux
tmux
sudo apt update && sudo apt upgrade
raspi-config
Raspberry Pi 3 Model B Plus Rev 1.3
2 Network Options N1 Hostname: rpi19
2 Network Options N2 Wi-fi: CZ Czech Republic
2 Network Options N2 Wi-fi: SSID PJN
2 Network Options N2 Wi-fi: Passphrase: ****
2 Network Options N3 Network interface names: enable
3 Boot Options B1 Desktop / CLI B1 Console:
    Text console, requiring user to login

```

```

3 Boot Options B2 Wait for Network at Boot: Yes
4 Localisation Options I1 Change Locale: cs_CZ.UTF-8 (en_GB.UTF-8 UTF-8)
4 Localisation Options I2 Change Timezone: Europe / Prague
8 Update:
raspi-config je již nejnovější verze (20191210).
apt install vim
echo "set -o vi" >> /etc/bash.bashrc
vim /etc/ssh/sshd_config
#PermitRootLogin prohibit-password
PasswordAuthentication no
useradd -c "Jiri Navratil" -m -g users -G sudo -s /bin/bash -u 5006 \
    navratil
echo "navratil ALL=(ALL) NOPASSWD: ALL" > \
    /etc/sudoers.d/020_navratil-nopasswd
cat >> ~/.ssh/authorized_keys << "EOF"
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIM615yS+/5KditOmGtI+1FzAFTO
    umcI2Qb37I+dERVVn jiri@navratil.cz
EOF
reboot

```

Přihlášení pomocí ssh klíče pod nového uživatele a smazání původního uživatele pi.

```

ssh navratil@192.168.4.83
sudo userdel -r pi
rm /etc/sudoers.d/010_pi-nopasswd

```

Zdrojový kód `rh` jsem na stroj `rpi19.navratil.lan` přenesl pomocí Gitu. Pro doplnění vypisování kromě terminálu také o výstup do console na LCD displej jsem potřeboval vypnout používání bufferu v Pythonu. Jedna z možností byla doplnit první řádek hlavního modulu o parametr `-u1`, tj. řádek by vypadal takto.

```
#! /usr/bin/env python3 -u
```

V kódu jsem však používal příkaz `print` pouze na dvou místech. Jeden pro tisk hlavičky a druhý pro informační řádek. Využil jsem tedy možnosti o doplnění těchto dvou příkazů o

```
, flush=True)
```

Dále jsem změnil způsob volání aplikace `rh`. Výstup z aplikace je v jednom okně s využitím roury a příkazu `tee` připisován na konec textového souboru.

```
./rh.py | tee -a /tmp/rh.txt
```

V jiném okně s právy řadového uživatele program `tail` tento textový soubor čte a posílá ho přes rouru na znovu na program `tee`, ale tentokrát s právy superuživatele a se zapisováním na zařízení `/dev/console`.

```
tail -f /tmp/rh.txt | sudo tee /dev/console
```

Tímto způsobem bylo možné s drobným zásahem do aplikace a s upraveným spuštěním zajistit, aby osoby děti a ženy měly 24x7 přístup k informacím z aplikace `rh`. LCD displej nevyžadoval zapojení silnějšího zdroje. Raspberry Pi je tichý a displej příliš nevyzařuje, takže jeho umístění v obyváku, který je současně ložnicí, není rušivé.

Na obrázku 4.8 je vidět `rpi19.navratil.lan` s běžící aplikací `rh` a výstupem zobrazovaným na LCD displeji.

<sup>1</sup> Vypíná buffer pro stdout a stderr



Obrázek 4.8. Raspberry Pi s LCD displejem

## 4.7 PostgreSQL

Průběžné ukládání vybraných dat ze Z-Wave zařízení do InfluxDB je užitečné k odlehčení zátěže Z-Wave sítě a otevírá možnost analyzovat časové řady uložených údajů. Zapisování výstupů z řídicí aplikace `rh` do textového souboru poskytuje log a dohledávání informací, které byly zobrazeny na terminálu a uloženy do textového souboru. Zajímaly mne hodnoty jako maximální denní výroba, které sice šly dohledat v logu, ale nebylo to přímočaré a rychlé. Rozhodl jsem se tedy aplikaci doplnit o sledování denních extrémů.

V domácí síti provozuji PostgreSQL server, který se na ukládání takových dat hodí.

PostgreSQL mi běží na Raspberry Pi `rpi03.navratil.lan` s IPv4 `192.168.4.52`. Mám na něm administrátorský účet. Vytvořil jsem novou databázi `rh`.

```
CREATE DATABASE rh WITH OWNER = navratil ENCODING = "UTF-8";
```

Na `rpi19.navratil.lan` jsem nainstaloval potřebné balíčky.

```
sudo apt install libpq-dev postgresql-client
```

Připojil jsem se na `rpi03` a otestoval, že mám časovou zónu nastavenou na Evropa/Praha nikoli na UTC.

```
psql -h 192.168.4.52 -U navratil -d rh
psql (11.9 (Raspbian 11.9-0+deb10u1))
SSL spojení (protokol: TLSv1.3, šifra: TLS_AES_256_GCM_SHA384, bitů: 256,
komprese: vypnuto)

rh=# SHOW TIMEZONE;
    TimeZone
-----
Europe/Prague
```

Následně jsem vytvořil jsem potřebné tabulky. Níže je tabulka pro denní maxima výroby z FVE a tabulka pro ukládání kompletní dat ze střídače ve formě JSON. Kompletní `schema.sql` použité k vytvoření tabulek je v příloze G.



```

CREATE TABLE IF NOT EXISTS vyroba_max (
    id SERIAL PRIMARY KEY,
    mtime TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP NOT NULL,
    hodnota SMALLINT DEFAULT 0
);
COMMENT ON TABLE vyroba_max IS 'denní maximální výroba z FVE';

CREATE TABLE IF NOT EXISTS stridac (
    id SERIAL PRIMARY KEY,
    mtime TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP NOT NULL,
    power_flow_realtime_data JSONB NOT NULL
    -- GetPowerFlowRealtimeData
);
COMMENT ON TABLE stridac IS 'JSON pro střídač';

```

Dále jsem vytvořil účet `script01` pro přístup z aplikace do `rh` databáze.

```

CREATE ROLE script LOGIN;
CREATE USER script01 WITH PASSWORD 'heslo' IN ROLE script;

```

a upravil jsem přístup na úrovni sítě.

```

root@rpi03:~# grep script01 /etc/postgresql/11/main/pg_hba.conf
host    rh                script01          192.168.4.0/24      md5
systemctl restart postgresql

```

Do aplikace `rh` jsem přidal modul `postgres` umožňující připojení na PostgreSQL RDBMS. U všech tabulek pro sledování jedné hodnoty jsem použil stejné označení. V modulu jsem potom implementoval metodu `save(self, table, value)`, která má do zvolené tabulky a jejího sloupce nazvaného vždy `hodnota` obsah z `value`. Tato metoda je zajímavá tím, že se nejprve pokusí aktualizovat hodnotu, pokud už pod stejným dnem existuje. Pokud operace `update` neuspěje, tak se vloží jako nová hodnota.

```

def save(self, table, value):
    updated_rows = inserted_rows = 0
    sql_update = """UPDATE {}
SET hodnota = %s, mtime=CURRENT_TIMESTAMP
WHERE DATE(mtime) = DATE(now())"""
    sql_insert = """INSERT INTO {}
(hodnota) VALUES (%s)"""
    t = (value, )
    query = psycopg2.sql.SQL(sql_update).format(
        psycopg2.sql.Identifier(table)
    )
    if DEBUG:
        print(query.as_string(self._conn))
    self._cur.execute(query, t)
    updated_rows = self._cur.rowcount
    if not updated_rows:
        query = psycopg2.sql.SQL(sql_insert).format(
            psycopg2.sql.Identifier(table)
        )
        if DEBUG:
            print(query.as_string(self._conn))
        self._cur.execute(query, t)
        inserted_rows = self._cur.rowcount

```

```

if DEBUG:
    print("Rows updated {} / inserted {}".format(
        updated_rows, inserted_rows)
    )
    print(sql_update)
    print(sql_insert)
    print(t)
self._conn.commit()

```

Metoda `save(self, table, value)` je potom použita v modulu `inverter` například pro uložení maximální výroby.

```

if self.P_PV > self.P_PV_highest:
    self.P_PV_highest = self.P_PV
    self._rdb.save("vyroba_max", self.P_PV)

```

Další důležitá metoda je `save_power_flow_realtime_data(self, data)`, která ukládá celý JSON ze střídače.

```

def save_power_flow_realtime_data(self, data):
    inserted_rows = 0
    sql_insert = """INSERT INTO stridac
(power_flow_realtime_data) VALUES (%s)"""
    t = [json.dumps(data)]
    self._cur.execute(sql_insert, t)
    inserted_rows = self._cur.rowcount
    if DEBUG:
        print("Rows inserted {}".format(inserted_rows))
        print(sql_insert)
        print(t)
    self._conn.commit()

```

Metoda `save_power_flow_realtime_data(self, data)` je potom použita v modulu `inverter` pro uložení kompletního JSON ze střídače.

```

r = requests.get(self.url, timeout=self._timeout)
r.raise_for_status()
powerflow = r.json()
self._rdb.save_power_flow_realtime_data(powerflow)

```

Náhodně vybraný JSON objekt z databáze. Formátování výstupu ručně upraveno pro lepší čitelnost.

```

SELECT * FROM stridac ORDER BY RANDOM() LIMIT 1;
id: 8395317
mtime: 2020-07-18 16:12:11.644717+02
power_flow_realtime_data:
{ "Body": {
  "Data": {
    "Site": {
      "Mode": "bidirectional",
      "P_PV": 3885,
      "E_Day": 12494,
      "E_Year": 3836110.4000000004,
      "P_Akku": 26.12,
      "P_Grid": -1342.23,
      "P_Load": -2409.77,

```

```

        "E_Total": 20632897.400000002,
        "BackupMode": false,
        "rel_Autonomy": 100,
        "BatteryStandby": false,
        "Meter_Location": "grid",
        "rel_SelfConsumption": 64.22627931769723
    },
    "Version": "12",
    "Inverters": {
        "1": {
            "P": 3752,
            "DT": 99,
            "SOC": 99,
            "E_Day": 12494,
            "E_Year": 3836110.5,
            "E_Total": 20632898,
            "Battery_Mode": "normal"
        }
    }
}, "Head": {
    "Status": {
        "Code": 0,
        "Reason": "",
        "UserMessage": ""
    },
    "Timestamp": "2020-07-18T16:12:09+02:00",
    "RequestArguments": {}
}}

```

Maximálních pět dodávek do sítě v roce 2020.

```

SELECT mtime, hodnota FROM do_site_max ORDER BY hodnota DESC LIMIT 5;
-----+-----
2020-11-21 14:27:45.62236+01 | 4889
2020-10-12 12:16:33.260386+02 | 4870
2020-11-06 11:43:44.69091+01 | 4863
2020-10-08 15:50:58.909022+02 | 4844
2020-10-11 15:36:32.624034+02 | 4825

```

Maximální naměřené hodnoty výroby v roce 2020 jsou v tabulce 4.1 a více hodnot v příloze I.

## 4.8 Zálohování a archivace

Pro zálohování Z-Way serveru je potřeba použít dvě různé zálohy. Přes `http://hub:8083/smarthome/#/admin` a volbu Backup & Restore se vygeneruje

```
512K    z-way-backup-2018-03-25-07-44.zab
```

a dále přes `http://hub:8083/expert#/network/control` → `http://hub:8083/ZWaveAPI/Backup` se vygeneruje

```
224K    z-way-backup-2018-03-25-07-46.zbk
```

maximální výroba	kWh
6.6.2020 12:27:18	8045
26.5.2020 12:33:20	7873
2.5.2020 13:49:22	7860
25.5.2020 12:41:15	7839
9.7.2020 12:43:57	7724
22.5.2020 12:15:41	7681
6.7.2020 12:26:25	7566
6.5.2020 12:04:49	7560
18.7.2020 11:00:21	7491
28.5.2020 14:07:26	7479
25.4.2020 11:48:33	7444
27.8.2020 11:52:33	7313
3.5.2020 10:54:44	7284
6.5.2020 12:52:57	7181
17.7.2020 11:40:31	7143
25.6.2020 12:35:35	7138
14.5.2020 12:50:18	7108
14.4.2020 12:03:28	7078
3.7.2020 13:38:57	7078
23.5.2020 12:59:50	6990

**Tabulka 4.1.** Maximální naměřené hodnoty výroby v roce 2020.

Pro zálohování PostgreSQL databáze ze stroje `rpi03.navratil.lan`.

```
PGPASSWORD=heslo pg_dump -h 192.168.4.52 -U navratil -d rh > \
  rpi03-rh-dump-$(date +%Y%m%d').sql
du -sh rpi03-rh-dump-20210101.sql
8.4G    rpi03-rh-dump-20210101.sql
lz4 -z9 --rm rpi03-rh-dump-20210101.sql
du -sh rpi03-rh-dump-20210101.sql.lz4
739M    rpi03-rh-dump-20210101.sql.lz4
```

Zálohování Git repozitáře na stroji `ohtar.navratil.lan`.

```
cd ~/src/rh
git bundle create /tmp/ohtar-rh-$(date +%Y%m%d').gitbundle --all
Enumerating objects: 572, done.
Counting objects: 100% (572/572), done.
Compressing objects: 100% (474/474), done.
Total 572 (delta 349), reused 162 (delta 89), pack-reused 0
du -sh /tmp/ohtar-rh-20210329.gitbundle
6.7M    /tmp/ohtar-rh-20210329.gitbundle
```

Zálohování Raspberry Pi počítačů na OpenBSD server `gildor.navratil.lan`. Pří-  
klad pro `rpi04.navratil.lan`.

```
cat zaloha-rpi04-rsync.sh
d=$(date +%Y%m%d')
FOLDER=/mnt/JN2TB05/rpi04
COMPUTERNAME=rpi04.navratil.lan
doas rsync -avz -e ssh --delete-after --delete-excluded \
  --progress --stats \
```

```

--exclude="/tmp/**" --exclude="/mnt/**" --exclude="/sys/**" \
--exclude="/var/run/**" \
--exclude="/proc/**" \
--exclude="/run/**" \
root@192.168.4.9:/ $FOLDER/rsync
echo datum $d
echo "cd $FOLDER/rsync &&"
echo "doas gtar -cpf ../$COMPUTERNAME-$d.gtar ."
echo "doas chown navratil ../$COMPUTERNAME-$d.gtar"

```

Zašifování před přesunutím do jiné lokality.

```

for file in *.dump *.tar *.gtar; do
  # do it only if file exist
  if [[ -e "$file" ]] ; then
    echo ${file};
    sha256 $file > $file.SHA256;
    lz4 -z9 --rm $file;
    ccrypt -e -k ~/.password ${file}.lz4;
  fi
done

```

## 4.9 Dokumentace

Zdrojový kód aplikace RaZberry Home a provozní dokumentace v MkDocs byla náhrana jako příloha `rh-src_and_mkdocs.tar` dne 28. března 2021 do Informačního systému na <https://is.czu.cz/>.

Kontrolní informace o souboru.

```

ls -l rh-src_and_mkdocs.tar
-rw-r--r-- 1 navratil users 1904640 bře 28 22:59 rh-src_and_mkdocs.tar
du -sh rh-src_and_mkdocs.tar
1,9M    rh-src_and_mkdocs.tar
sha256sum rh-src_and_mkdocs.tar
f4c17f3b54786f86c436aa5edfd298c4091b05ad79d1e88acf1f80b9e0cb600a
rh-src_and_mkdocs.tar

```

Rozbalení `rh-src_and_mkdocs.tar` a spuštění MkDocs v režimu poskytování dokumentace na lokálním hostu na portu 8000.

```

tar xf rh-src_and_mkdocs.tar
cd rh
mkdocs serve

```

Pokud máte MkDocs, tak příkaz `mkdocs serve` umožní prohlížení dokumentace ve webovém prohlížeči na `http://127.0.0.1:8000`. Jinak je možné dokumentaci prohlížet otevíráním vygenerovaných `index.html` souborů nebo čtením Markdown souborů v docs adresáři. Adresář `docs` obsahuje zdroj dokumentace a adresář `site` obsahuje dokumentaci z docs vygenerovanou.

Obsah tar souboru filtrovaný na zdrojový kód aplikace a provozní dokumentaci.

```

tar tf rh-src_and_mkdocs.tar | sort | grep -v site
rh/
rh/config.py
rh/device.py

```

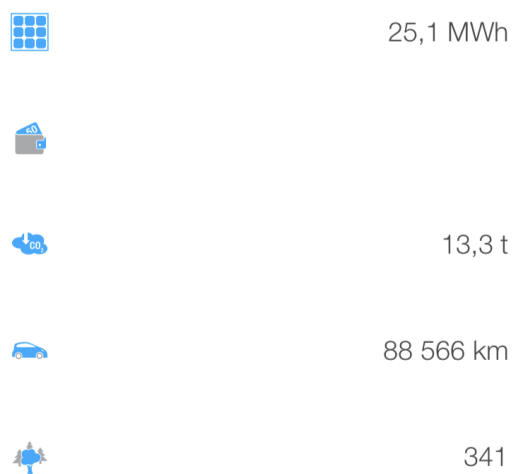
```
rh/docs/  
rh/docs/about.md  
rh/docs/fronius.md  
rh/docs/index.md  
rh/docs/mkdocs.md  
rh/docs/pocitace/  
rh/docs/pocitace/rpi03.md  
rh/docs/pocitace/rpi04.md  
rh/docs/pocitace/rpi19.md  
rh/docs/ucel.md  
rh/.gitignore  
rh/hub.py  
rh/INSTALL.md  
rh/inverter.py  
rh/mkdocs.yml  
rh/postgres.py  
rh/README.md  
rh/requirements.txt  
rh/rh.py  
rh/room.py
```

Obsah tar souboru filtrovaný na vygenerované index.html soubory provozní dokumentace.

```
tar tf rh-src_and_mkdocs.tar | sort | grep index.html  
rh/site/about/index.html  
rh/site/fronius/index.html  
rh/site/index.html  
rh/site/mkdocs/index.html  
rh/site/pocitace/rpi03/index.html  
rh/site/pocitace/rpi04/index.html  
rh/site/pocitace/rpi19/index.html  
rh/site/ucel/index.html
```

## 4.10 Provozní údaje FVE

Fotovoltaická elektrárna byla v provozu od 21.04.2017 12:39 do 14.12.2020 09:13 a za tu dobu vyrobila 25,1 MWh elektrické energie což odpovídá úspoře 13,3t  $CO_2$ , ujetých 88 566 km po dálnici nebo zasazení 341 stromů (obrázek 4.9). Další údaje o provozu a informace o tom, proč už v provozu není najdete v příloze C.



Poslední aktualizace: 14.12.2020 9:10

**Obrázek 4.9.** Celkově vyrobená elektrická energie za dobu provozu

## Kapitola 5

### Zhodnocení výsledků a doporučení

Nadchla mne možnost spojit řešení mé reálné situace se závěrečnou prací. Na jedné straně bylo výhodou, že jsem si mohl řešení doslova ošahat a změny byly ihned vidět. Na druhou stranu pro mne bylo těžké udržet při psaní odbornou rovinu. Měl jsem potřebu psát o mém osobním příběhu. Některé části jsem několikrát přepsal, ale uvědomuji si, že jsem nebyl schopen práci překlopit do čistě odborného textu. Mám dobrý pocit z toho, jak jsem v práci zvládl psaní kódu. Na psaní odborného textu mám co zlepšovat.

Za dobu tvoření práce jsem vygeneroval více poznámek a logů, než jsem byl schopen do tohoto textu dostat. Kvůli udržení limitu na počet stran jsem vytvořil provozní dokumentaci v MkDocs a spolu s kompletním zdrojovým kódem jsem ji odevzdal jako přílohu. V rámci vkládání logů a částí kódu do práce jsem se snažil u logů poskytnout dostatek informací, aby byl postup zachycen kompletně. U zdrojových kódů jsem vytáhl z aplikace podstatné části a popsal jsem je tak, abych čtenáři ulehčil pronikání do kódu, který je součástí přílohy.

Závěrečný rok práce byl rokem pandemickým. Infrastruktura v domě ustála někdy až pět videostreamů současně. Svůj podíl na tom mohla mít od Wi-Fi oddělená chytrá domácnost provozovaná v síti Z-Wave. Zajímavé v této souvislosti bylo zjištění Patrika Zandla, že problém jeho domácí Wi-Fi byl vysavač a žárovka s Wi-Fi [41].

Home office a studium z domova vyžadovalo během topné sezóny upravení režimu vytápění. Aplikace byla již na takové úpravy připravená, takže stačilo přidat dny pro home office a vytápění se chovalo, jak jsme potřebovali.

Přidání ochrany proti zamrznutí mi umožnilo opouštět dům na delší dobu, aniž bych se musel obávat promrznutí domu.

Velmi nepříjemná byla zkušenost s poškozenou střechou. Závada z chybné instalace se projevila teprve po třech letech provozu během bouřky a extrémních poryvů větru. Po několika pokusech o provizorní opravu střechy musela být nakonec fotovoltaická elektrárna demontována. Naštěstí pro mne se PRE k problému postavilo čelem. To samé se bohužel nedá říct o SFŽP ČR. Naše interpretace podmínek dotace se zásadně liší. Já zastávám stanovisko, že jsem měl být závazku udržet fotovoltaickou elektrárnu v provozu 10 let zproštěn na základě budu 3.6 podmínek pojednávajícím o nepředvídatelné události nebo se měl přinejmenším nárok na dotaci přepočítat dle bodu 9.1 odstavce 8 na 94 125 Kč ze 150 000 Kč obdržených dle vyrobených 25,1 MWh. Doufám, že v okamžiku obhajoby této práce již budu vědět, zda bude SFŽP ČR požadovat vrácení dotace v plné výši, nebo v poměrné výši 55 875 Kč za „nevyrobenou energii“ nebo akceptuje neporušení podmínek dle bodu 3.6 o nepředvídatelné události.

Oblast podpory a nepodpory zelené energie by zasloužila samostatnou práci. Ceny za výkup vyrobené zelené energie od soukromých osob bez licence pouze lehce převyšují poplatky, které si například BOHEMIA ENERGY v mém případě účtuje viz příloha E.13. Přitom při prodeji zelené energie si nechávají dodavatelé za zelenou energii připlatit. Blíží se nové dotace pro obnovitelné zdroje od Modernizačního fondu, který



---

čerpá prostředky z prodeje emisních povolenek. Bude zajímavé pozorovat, jakým způsobem to ovlivní rozvoj fotovoltaiky v České republice.

Počítač Raspberry Pi, se kterým jsem měl již zkušenost před nasazením v této práci, se v rámci této práce jednoznačně osvědčil. Malý, tichý s nízkou spotřebou, dostatečným výkonem a s rozhraním GPIO, které umožnilo připojení modulu pro chytrou domácnost a také připojení LCD displeje.

Nasazení druhého Raspberry Pi počítače s LCD displejem zjednodušilo přehled nejenom pro ostatní členy domácnosti, ale nakonec překvapivě také pro mne. Přestože mi nedělá problém se přes ssh na řídicí počítač připojit a podívat se na aktuální stav domácnosti, tak zajít do obýváku a kouknout na displej je rychlé a pohodlné.

Způsob rozšíření aplikace o vypisování do console byl překvapivě jednoduchý a přitom ponechal počítač zabezpečený a bez potřeby autentizace uživatele, který chce pouze pozorovat stav.

Po psaní bakalářské práce v aplikaci MS Word jsem vděčný za možnost použít pro diplomovou práci  $\text{\TeX}$ . Sice jsem musel na začátku věnovat čas úpravě šablony dle ČZU grafického manuálu, ale následně jsem nechal díky sázení v  $\text{\TeX}$ u typografickou stránku na Danaldu Knuthovi a panu Olšákovi a mohl jsem se soustředit na obsah.

Při tvorbě obsahu jsem si byl vědom, že je práce velmi prakticky zaměřená, proto jsem se snažil co nejvíce nastudovat odborné materiály, aby byla v práci dostatečně zastoupena i teorie. S blížícím se termínem odevzdání práce jsem si uvědomil, že přestože mám fyzickou a programovou stránku řešení hotovou řadu měsíců, tak mám tendenci studovat další a další odborné materiály a málo píšu. Beru si z toho poučení, že je potřeba začít psát co nejdříve.

Možnost připojení střídače Fronius do sítě LAN a dostupnost API rozhraní bylo pro úspěch práce klíčové. Bylo doslova štěstí, že součástí dodávky fotovoltaické elektrárny byl takto vybavený střídač.

Koncepce autonomního režimu bez potřeby cloudových služeb se osvědčila a funguje. Chod domu mám plně pod kontrolou a nezávislé na externích firmách a jejich službách. Dále poskytuje řešení ukryté za firewallem bez možnosti adresování chytrých zařízení a bez ukládání jejich stavu mimo dům vyšší bezpečnost a vyšší soukromí.

Již při návrhu jsem myslel na to, že nechci přetěžovat zařízení napájená z monočlánků a nasadil jsem proto databázi pro časové řady pro snížení počtu dotazů na chytrá zařízení. Přesto mne překvapilo, že je v zařízeních nutné měnit monočlánky 2-3 za rok. Pokud jsem monočlánky nevyměnil v předstihu před úplným vybitím, tak se průchodnost sítě snížila a někdy bylo nutno dokonce zařízení znovu připojovat do sítě.

Reálný dosah Z-Wave zařízení byl v mých podmínkách nižší než teoretické hodnoty. Pro stabilní síť jsem proto po domě rozmístil i 220V Z-Wave zásuvky, které slouží výhradně k lepšímu šíření dat po Z-Wave síti.

Díky aktuálním datům o toku elektrické energie v domácnosti jsem například zjistil, že moje elektrická indukční plotýnka upravuje výkon nelineárně, ale pouze prodlužováním či zkracováním času zapnutí. Na střídači bylo vidět, jak musí stále vyrovnávat výkyvy v odběru. Představa většího množství spotřebičů, které takto zatěžují rozvodnou soustavu, mi zní přinejmenším jako nepříjemnost.

Teprve jako uživatel fotovoltaické elektrárny jsem pochopil, jak je v zimním období i za dne plného slunce vyrobeno málo elektřiny. Zjednodušeně se dá napsat, že pět měsíců v roce se mi nedařilo dosahovat soběstačnosti.

Dobrá zpráva je, že mimo těchto pět měsíců byl dům převážně soběstačný. Spotřebu jsem řídil tak, abych přečkal noc „na baterku“ a přitom jsem vyrobenou energii využíval na maximum. Z porovnání obrázku D.11 a obrázku D.12 je patrné, že v roce 2020

oproti roku 2019 odevzdaná energie klesala, energie uložená v akumulátoru stoupala a energie spotřebovaná stoupala. To byl můj záměr a cíl této práce. Dosáhl jsem toho díky nasazené automatizaci.

Chtěl bych zpracovat a vytěžit z dat uložených v PostgreSQL a v InfluxDB další zajímavé informace. V příloze I jsou z PostgreSQL vtaženy záznamy o maximální výrobě v roce 2020. Na těchto datech je ukázáno, ve kterých dnech a denní době roku 2020 bylo dosaženo maximálního výkonu fotovoltaické elektrárny a jakého výkonu bylo dosaženo. Současně se tím prokázalo, že panely byly schopny vyrobit více, než kolik udávají jejich tabulkové hodnoty. Jako velmi cenné se teď s odstupem 15 měsíců a za stavu demonstované fotovoltaické elektrárny ukázalo rozhodnutí do PostgreSQL ukládat kompletní provoz střídače v JSON formátu. S těmito daty jsem schopen z uložených dat zpětně vytěžit i informace, které jsem neukládal do vyčleněných tabulek.

Zvažoval jsem, zda budu schopen udělat nějaké statistické závěry. Bohužel pro statistické či dokonce prognostické závěry nebyla fotovoltaická elektrárna v provozu dostatečně dlouho. Navíc, jak i sám ERÚ ve svých materiálech uvádí, na spotřebu elektrické energie má velký vliv chladné počasí. Já jsem z provozu fotovoltaické elektrárny vyzoroval, že v rámci roční doby je možné sezónnost zachytit, ale počasí má stále velký vliv na množství vyrobené energie. Jako ukázka může být porovnán červen roku 2019 a červen roku 2020. Na obrázku D.11 je vidět v roce 2019 vyšší výroba v červnu oproti květnu i červenci. Naopak na obrázku D.12 je vidět v roce 2020 nižší výroba v červnu oproti květnu i červenci. Podobná situace s rokem 2019 byla v roce 2017 a podobná situace s rokem 2020 byla v roce 2018. Rozhodně v tom nespátřuji korelaci v naměřených hodnotách u lichých resp. sudých let. Na naměřených datech chci ukázat, že počasí má na výrobu zásadní vliv.

Mezi nápady, které bych rád v budoucna zrealizoval, je sledování počasí a energie potřebné na vytopení domu. Mezi zajímavá zjištění řadím zaznamenání toho, že z 21 °C na 16 °C klesne teplota v zimním období při vypnutém vytápění mnohem rychleji než ze 16 °C na 11 °C. Podobně udržení teploty v domě na 21 °C vyžaduje nezanedbatelně více energie než její udržení na 20 °C. Proto mezi jednou z důležitých úprav přínášející úsporu bylo zavedení různých teplotních rozsahů vytápění. Jméno proměnné `self._thermal_comfort_for_women` rozlišující zvolený režim vytápění by měla být sebe vysvětlující.

Několikrát jsem využil ostrovní režim viz obrázek 5.1 a pokaždé jsme byli schopni v domácnosti fungovat. Narazil jsem při tom na dva limity. První byl nedostupnost Wi-Fi připojení do Internetu, protože při výpadky větší části Suchdola neměl přístupový bod na který se připojují elektrickou energii a tudíž byl pro mne nedostupný<sup>1</sup>. Druhý byl nemožnost použití spotřebiče na více fázích jako indukční plotýnka a varná konvice.

---

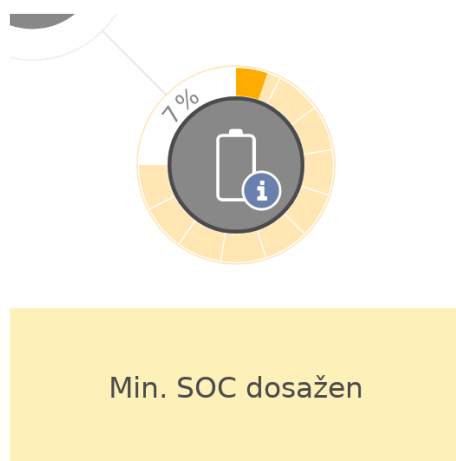
<sup>1</sup> Jednou jsem použil mobilní síť, která fungovala.

## Zprávy

#	Název přístroje	Datum
	System Monitoring Leave backup mode	30.07.2020 11:26
989	System Monitoring No Smart Meter communication	30.07.2020 10:45
988	System Monitoring Smart Meter communication error	30.07.2020 10:44
	System Monitoring Enter backup mode	30.07.2020 10:44

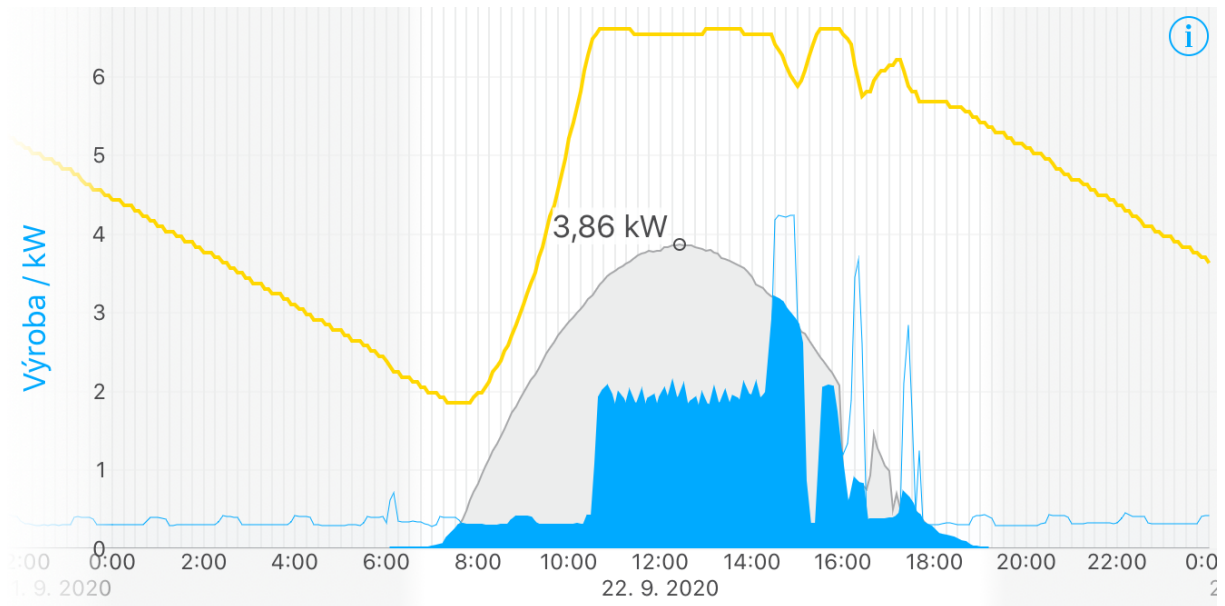
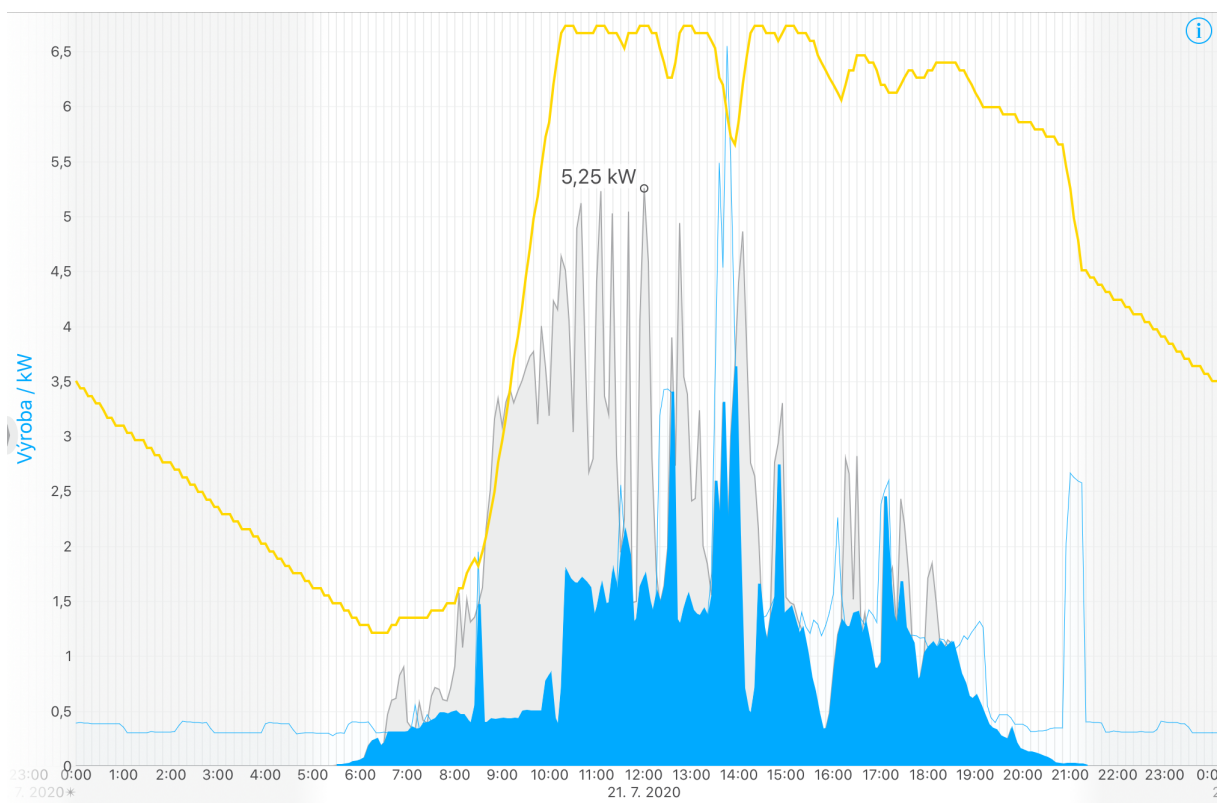
**Obrázek 5.1.** Ostrovní režim 30. 7. 2020

Další nová zkušenost byla s baterií. V zimním období se bohužel moc nenabíjela. Přestože střídač hlídal, aby se nevybíjela pod 7% viz obrázek 5.2, tak se stalo, že baterka „úplně umřela“ a přestala komunikovat. Po této zkušenosti jsem upravoval v zimě manuálně nastavení minimálního nabití na 30%.



**Obrázek 5.2.** Dosažení minimálního povoleného nabití akumulátorů

Účinnost automatizovaného režimu je dobře viditelná na obrázku 5.3 a na obrázku 5.4. V obou případech baterka přečká noc (žlutá čára), v době dostatečného nabití baterky a dostatečného výkonu solárních panelů se zvyšuje spotřeba (zapnutím odvlhčovačů) a při poklesu jsou vypnuty. Obrázek 5.3 je ze slunného dne, obrázek 5.4 ze dne s mraky, ale průběh je podobný.

**Obrázek 5.3.** Výroba a spotřeba dne 22. 9. 2020**Obrázek 5.4.** Výroba a spotřeba dne 21. 7. 2020

Pro případnou snahu předvídat výrobu v rámci dne může být užitečná informace u iOS aplikace CHMU, která udává hustotu mraků. V kombinaci s ročním obdobím a denní dobou se dá díky této informaci odhadnout, jak se bude výroba elektřiny vyvíjet.



**Obrázek 5.5.** Hustota mraků dle iOS aplikace CHMU

Z provozu mé fotovoltaické elektrárny je pro výrobu elektřiny nejlepší období od dubna do září. Z pohledu náklady za elektřiny nemám bohužel řádné podklady pro porovnání. Měnil jsem dodavatele energií, pandemie nás drží doma a chladná období se mění.

V roce 2017 jsem při hledání řídicí jednotky připojitelné na Raspberry Pi přes GPIO nenašel žádnou pro technologii Zigbee. Situace se změnila a v roce 2020 byla k dispozici <https://phoscon.de/en/raspbee> a nyní v březnu 2021 je možné koupit <https://phoscon.de/en/raspbee2>. IKEA a Lidl prodávají Zigbee zařízení. Zvážím, zda nepřidat do řešení Zigbee na dalším Raspberry Pi a doplním domácnost o další chytrá zařízení.

## Závěr

Provozování fotovoltaické elektrárny a vybudování chytré domácnosti mne bavilo a mnoho jsem se naučil.

Za pomoci minipočítače Raspberry Pi 3 se mi podařilo cíl i podcíl práce splnit. Jeho výkon je dostatečný a spolu s dostupností software a rozšiřujícího hardware bylo možné vybudovat autonomní systém na monitorování výroby elektrické energie z fotovoltaické elektrárny, k monitorování stavu nabití akumulátorů a k regulaci spotřeby elektrické energie.

Realizace byla rozšířena o LCD displej zobrazující aktuální stav pro všechny členy domácnosti. Byl k tomu použit druhý<sup>1</sup> stroj Raspberry Pi 3 a byl na něho přesunut řídicí program.

Raspberry Pi počítače jsou malé, tiché a mají nízkou spotřebu. Raspberry Pi s LCD displejem nepůsobí nijak rušivě. Provozují ho v obýváku, který je současně ložnicí a jeho provoz nás nijak neobtěžuje.

Použití Z-Wave technologie mi umožnilo realizovat chytrou domácnost odděleně od vnitřní Wi-Fi sítě. Během pandemického roku musela Wi-Fi síť občas zvládnout až pět paralelních videostreamů. Věřím, že nezatěžování Wi-Fi sítě čtrnácti zařízeními chytré domácnosti k spolehlivému fungování sítě Wi-Fi pomohlo.

Do budoucna se nabízí nasazení dalšího Raspberry Pi počítače pro doplnění chytré domácnosti o Zigbee technologii. Nově je možné Raspberry Pi rozhraní GPIO osadit řídicí jednotkou pro Zigbee technologii nazvanou RaspBee II. V obchodech jako IKEA a LIDL jsou k dostání zařízení pro chytrou domácnost podporující Zigbee. Po ověření, že se nebudou technologie na fyzikální úrovni vzájemně rušit, mohu přidat Zigbee jako další část chytré domácnosti. Rozšíření řídicí aplikace o Zigbee by mělo být realizovatelné a podobné nasazení Z-Wave, ale i to musím nejprve ověřit.

Data z chodu domácnosti byla zaznamenána do InfluxDB a do PostgreSQL. Některá data jsem již použil v této práci a další data čekají na zpracování. Za velmi šťastné rozhodnutí považuji ukládání JSON údajů ze střídače do PostgreSQL. Díky tomu je možné zpětně zrekonstruovat provoz, dohledat libovolný údaj ze střídače a dokonce mohu uložená data použít na simulaci provozu.

Pro tuto práci jsem v průběhu plánování i realizace řešení vytvořil velké množství textu, které se podle plánu stalo podkladem pro tento text práce. Tím, že byly podklady tvořeny v různém čase a v různých situacích, došlo bohužel k tomu, že nebyly psány stejným stylem. Při jejich zapracování do textu, který právě čtete, bych odhadl s použitím Paretova pravidla, že 20% podkladů vytvořilo 80% textu práce a při jejich sladování a vylepšování jsem spotřeboval 80% času na změnu 20% textu.

Praktické pro mne bylo použití textového editoru Vim, psaní v TeXu Donalda Knutha a používání šablon od Petra Olšáka. Mohl jsem se soustředit na obsah a typografii jsem nechal na znalostech povolanejších.

Nalezení způsobu, jak zrcadlit výstup terminálové aplikace na displej počítače bez nutnosti přihlášení uživatele bylo užitečné v jednoduchosti a cenné v jeho obecnosti

<sup>1</sup> Potřebná GPIO pozice na prvním Raspberry Pi již byla obsazena RaZberry rozšířením.

---

a bezpečnosti použití. Kód má ošetřené výjimky, se kterými jsem se setkal za chodu či jsem je ideálně dopředu předpokládal. Testování v provozu umožnilo postupně narazit na různé krajní či neočekávané stavy a ošetřit je tak, aby aplikace běžela dál. Na hodnoty mimo očekávaný rozsah či na další atypické situace jsem použil lambda kalkul z funkcionálního programování. Osvědčila se mi konfigurace přes jeden konfigurační modul s využitím vlastností tříd. Pracoval jsem s množinami instancí a používal kromě metod instancí také metody tříd.

Budu potěšen, pokud jste v práci našli alespoň jednu informaci, která pro vás byla nová. Já jsem našel nových informací spoustu. Svá zjištění jsem zaznamenával do této práce a využíval při psaní kódu. Někdy pro mne bylo těžké udržet stranou emoce a nezdůraznit „wow“ efekt, který jsem při zjištění zažil. I jako 50letý si uvědomuji, že se mám stále co učit a chci v tom dále pokračovat. Proces učení a objevování nových věcí a poznatků mne stále baví.

Tato práce zodpověděla otázky položené na začátku i na řadu dalších. Stejně tak přinesla řadu otázek, které s moji prací blízce či vzdáleně souvisí. K některým otázkám se vrátím a budu na ně hledat odpovědi. Těším se na nalezení odpovědí a vlastně i na nové otázky, které mi život přinese.

*Jiří Navrátil*

## Literatura

- [1] PAETZ, Christian. *Z-Wave Basics: Remote Control in Smart Homes*. London: CreateSpace Independent Publishing Platform, 2013. ISBN 9781490537368.
- [2] *ITU Recommendation G.9959: Short range narrow-band digital radiocommunication transceivers - PHY, MAC, SAR and LLC layer specifications. 2015-01*. Geneva: International Telecommunication Union, 2015. Dostupné také z: <https://www.itu.int/rec/T-REC-G.9959>
- [3] Fronius Solar.web *Fronius International GmbH* [online]. 2020 [cit. 2021-01-02]. Dostupné z: <https://www.solarweb.com/>
- [4] Fronius Solar API V1 [online]. 13. 2020, 84 s. [cit. 2021-01-02]. IDENT 013-02062020. Dostupné z: <https://www.fronius.com/en/solar-energy/installers-partners/technical-data/all-products/system-monitoring/open-interfaces/fronius-solar-api-json->
- [5] Z-Way User and Programmer Manual [online]. 17.10.2017, 229 s. [cit. 2021-01-02]. Dostupné z: <https://z-wave.me/essentials> a z <https://github.com/Z-Wave-Me/Z-Way-Manual>
- [6] BUNZ, Mercedes a Graham MEIKLE. *The internet of things*. Cambridge: Polity, 2018. Digital media and society. ISBN 978-1-5095-1746-6.
- [7] NORRIS, Donald. *Raspberry Pi projects for the evil genius*. New York: McGraw Hill, 2014. Evil genius series (McGraw Hill). ISBN 978-0-07-182158-2.
- [8] NORRIS, Donald. *Raspberry Pi: projekty*. Brno: Computer Press, 2015. ISBN 9-788025-143469.
- [9] Energetický regulační úřad *Zprávy o provozu elektrizační soustavy* [online]. 2020 [cit. 2021-03-06]. Dostupné z: <http://www.eru.cz/cs/zpravy-o-provozu-elektrizacni-soustavy>
- [10] Ministerstva pro místní rozvoj *Fotovoltaika: metodická pomůcka Ministerstva pro místní rozvoj k umístování, povolování a užívání fotovoltaických staveb a zařízení* [online]. 2014 [cit. 2021-03-06]. Dostupné z: [https://www.mmr.cz/getmedia/b4a94988-32fd-4b69-a548-82a288467769/FVE-web\\_01\\_2014.pdf](https://www.mmr.cz/getmedia/b4a94988-32fd-4b69-a548-82a288467769/FVE-web_01_2014.pdf)
- [11] Zákon č. 131/2015 Sb.: *Zákon, kterým se mění zákon č. 458/2000 Sb., o podmínkách podnikání a o výkonu státní správy v energetických odvětvích a o změně některých zákonů (energetický zákon), ve znění pozdějších předpisů, a další související zákony*. [online] [cit. 2021-03-06]. Dostupné také z: <https://www.fulsoft-cz.infozdroje.czu.cz/a3d5p>
- [12] Zákon č. 183/2006 Sb.: *Zákon o územním plánování a stavebním řádu (stavební zákon)* [online] [cit. 2021-03-06]. Dostupné také z: <https://www.fulsoft-cz.infozdroje.czu.cz/a3d5q>
- [13] Zákon č. 458/2000 Sb.: *Zákon o podmínkách podnikání a o výkonu státní správy v energetických odvětvích a o změně některých zákonů (energetický zákon)* [online] [cit. 2021-03-06]. Dostupné také z: <https://www.fulsoft-cz.infozdroje.czu.cz/a3d5r>



- [14] Zákon č. 165/2012 Sb.: *Zákon o podporovaných zdrojích energie a o změně některých zákonů* [online] [cit. 2021-03-06]. Dostupné také z: <https://www.fulsoft.cz.infozdroje.czu.cz/a3d5o>
- [15] Energetický regulační věstník, částka 12/2016: *Cenové rozhodnutí Energetického regulačního úřadu č. 9/2016 ze dne 14. prosince 2016, kterým se mění cenové rozhodnutí ERÚ č. 5/2016, kterým se stanovuje podpora pro podporované zdroje energie* [online] [cit. 2021-01-02]. Dostupné z: <https://www.eru.cz/-/cenove-rozhodnuti-c-9-2016>
- [16] Energetický regulační věstník, částka 5/2020: *Cenové rozhodnutí Energetického regulačního úřadu č. 7/2020, ze dne 30. září 2020, kterým se stanovuje podpora pro podporované zdroje energie* [online] [cit. 2021-01-02]. Dostupné z: <https://www.eru.cz/-/cenove-rozhodnuti-c-7-2020>
- [17] E.ON Rádce *Co označuje jednotka kWp?* [online] [cit. 2021-01-02]. Dostupné z: <https://www.eon.cz/radce/chytra-domacnost/jak-vyuzivat-solarni-energii/co-oznacuje-jednotka-kwp>
- [18] Fronius *Fronius Symo Hybrid 5.0-3-S Datový list* [online] [cit. 2021-01-02]. Dostupné z: <https://www.fronius.com/cs-cz/czech-republic/solar-energy/vyroby/domacnosti/stridac/fronius-symo-hybrid/fronius-symo-hybrid-5-0-3-s>
- [19] Charity number: 1129409 *RASPBERRY PI FOUNDATION* [online] [cit. 2021-01-02]. Dostupné z: <https://register-of-charities.charitycommission.gov.uk/charity-search/-/charity-details/5002372/>
- [20] Eben Upton tweet *29.8 million unit sold* [online] [cit. 2021-01-02]. Dostupné z: <https://twitter.com/EbenUpton/status/1205906969443393537>
- [21] GitHub *Raspberry Pi revision codes* [online] [cit. 2021-01-02]. Dostupné z: <https://github.com/raspberrypi/documentation/blob/master/hardware/raspberrypi/revision-codes/README.md>
- [22] Směrnice rektora č. 6/2011 *o používání znaku ČZU a grafického manuálu ČZU* [online] [cit. 2017-09-21]. Dostupné z: <https://www.czu.cz/cs/r-7210-o-czu/r-7702-oficialni-dokumenty/r-7810-vnitri-predpisy-univerzity>
- [23] ČZU grafický manuál [online] [cit. 2017-09-21]. Dostupné z: <https://www.czu.cz/cs/r-7210-o-czu/r-7701-pr-a-media/r-8113-graficky-manual-a-loga>
- [24] ČZU Závěrečné práce *Požadavky pro předkládání diplomových prací PEF 2017* [online] [cit. 2018-10-21]. Dostupné z: <https://www.pef.czu.cz/cs/r-7008-studium/r-7026-informace-pro-studenty/r-12261-zaverecne-prace>
- [25] OpenBSD PF *Building a Router* [online] [cit. 2021-02-07]. Dostupné z: <https://www.openbsd.org/faq/pf/example1.html>
- [26] BERNSTEIN, Daniel J. a Niels DUIF a Tanja LANGE, a Peter SCHWABE a Bo-Yin YANG. *High-speed high-security signatures*. 2011 [online] [cit. 2021-01-12]. Dostupné z: <http://ed25519.cr.yip.to/ed25519-20110926.pdf>
- [27] GPIO *Raspberry Pi Documentation* [online] [cit. 2021-03-02]. Dostupné z: <https://www.raspberrypi.org/documentation/usage/gpio/>
- [28] Raspberry Pi 3 Model B *Specifikace* [online] [cit. 2020-11-27]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [29] Adafruit 954 Cable *USB to TTL Serial Cable* [online] [cit. 2020-08-12]. Dostupné z: <https://www.adafruit.com/product/954>
- [30] Z-Wave Alliance *Z-Wave Plus TM Certification* [online] [cit. 2021-01-04]. Dostupné z: <https://z-wavealliance.org/z-wave-plus-certification/>

- [31] Z-Wave.Me *RaZberry* [online] [cit. 2021-02-20]. Dostupné z: <https://z-wave.me/products/razberry/>
- [32] HIBBS, Curt, Steve JEWETT a Mike SULLIVAN. *The art of lean software development*. Sebastopol: O'Reilly, 2009). ISBN 978-0-596-51731-1.
- [33] Cucumber BDD Testing *Gherkin Reference* [online] [cit. 2021-02-20]. Dostupné z: <https://cucumber.io/docs/gherkin/reference/>
- [34] ČTK České noviny *Pražské střechy dají solární výkon 120.000 domácnostem* [online] [cit. 2021-03-24]. ISSN: 1213-5003 Dostupné z: <https://www.ceskenoviny.cz/zpravy/analyza-prazske-strechy-daji-solarni-vykon-120-000-domacnostem/2010458>
- [35] Seznam Zprávy *Soláry a větrníky zasype dotační smršť, jaká tu ještě nebyla* [online] [cit. 2021-03-24]. Dostupné z: <https://www.seznamzpravy.cz/clanek/solary-a-vetrniky-zasype-dotacni-smrst-jaka-tu-jeste-nebyla-hrozi-dalsi-bublina-147346>
- [36] MkDocs *Project documentation with Markdown*. [online] [cit. 2021-12-27]. Dostupné z: <https://www.mkdocs.org/>
- [37] Donald Ervin Knuth *Domovská stránka na Stnafordově univerzitě*. [online] [cit. 2021-03-24]. Dostupné z: <https://www-cs-faculty.stanford.edu/~knuth/>
- [38] TeX Users Group *History of TeX*. [online] [cit. 2021-03-24]. Dostupné z: <http://tug.org/whatis.html>
- [39] Petr Olšák *OPmac - rozšiřující makra plain TeXu*. [online] [cit. 2021-03-24]. Dostupné z: <http://petr.olsak.net/opmac.html>
- [40] OpenBSD *iostat.c* [online] [cit. 2018-02-13]. Dostupné z: <https://github.com/openbsd/src/blob/master/usr.sbin/iostat/iostat.c>
- [41] Patrik Zandl tweet *příčinou problémů domácího Wi-Fi byl vysavač a žárovka s Wi-Fi* [online] [cit. 2021-03-26]. Dostupné z: <https://twitter.com/tangero/status/1375546127668371456>

## Seznam použitých zkratek

V textu jsou použity tyto zkratky:

API	Application Programming Interface - aplikační rozhraní
BLE	Bluetooth Low Energy
CPU	central processing unit - centrální procesorová jednotka
DB	databáze
DSL	domain-specific language (DSL) - doménově specifický jazyk
ERÚ	Energetický regulační úřad
FVE	fotovoltaická elektrárna
GPIO	general-purpose input/output
IATA	International Air Transport Association
ICAO	Mezinárodní organizace pro civilní letectví
JSON	JavaScript Object Notation - zápis objektů v JavaScriptu
kWp	Kilowatt-peak je jednotka špičkového výkonu fotovoltaické elektrárny
LAN	Local Area Network - lokální síť
LKPR	ICAO označení pro Mezinárodní letiště Václava Havla Praha
MAC	Media Access Control - identifikátor síťového zařízení
MMR	Ministerstvo pro místní rozvoj
MPO	Ministerstvo průmyslu a obchodu
OTE	OTE, a.s. - operátor trhu na základě licence od ERÚ pro registrované účastníky a výrobce POZE
OZE	obnovitelné zdroje energie
POZE	podporované zdroje energie
RDBMS	Relational Database Management System - relační databázový systém
RPi	Raspberry Pi
PRG	IATA označení pro Mezinárodní letiště Václava Havla Praha
SFŽP ČR	Státní fond životního prostředí České republiky
SoC	system on a chip - systém na čipu
STC	Standard Test Conditions
UPS	Uninterruptible Power Source



## Přílohy

# Příloha A

## Parametry Fronius střídače

### TECHNICAL DATA FRONIUS SYMO HYBRID

/ The Fronius Symo Hybrid is the heart of the storage solution for 24H Sun - the Fronius Energy Package. From a simple inverter one minute, the battery and emergency power function can be added in no time. The result: sun by day, sun by night and sun during power outages.



INPUT DATA	SYMO HYBRID 3.0-3-S	SYMO HYBRID 4.0-3-S	SYMO HYBRID 5.0-3-S
PV input power	5.0 kW	6.5 kW	8.0 kW
Max. input current ( $I_{ik,max}$ )	1 x 16 A	1 x 16 A	1 x 16 A
Max. short circuit current, module array		24 A	
Min. input voltage ( $U_{dc,min}$ )		150 V	
Feed-in start voltage ( $U_{dc,start}$ )		200 V	
Nominal input voltage ( $U_{dc}$ )		595 V	
Max. input voltage ( $U_{dc,max}$ )		1000 V	
MPP voltage range ( $U_{mpp,min} - U_{mpp,max}$ )	200 - 800 V	255 - 800 V	320 - 800 V
Number of MPP trackers		1	
Number of DC connections (PV)		2	

BATTERY INPUT	SYMO HYBRID 3.0-3-S	SYMO HYBRID 4.0-3-S	SYMO HYBRID 5.0-3-S
Maximum output power to battery		Depends on connected Fronius Solar Battery	
Maximum input power from battery		Depends on connected Fronius Solar Battery	

OUTPUT DATA	SYMO HYBRID 3.0-3-S	SYMO HYBRID 4.0-3-S	SYMO HYBRID 5.0-3-S
AC nominal output ( $P_{ac}$ )	3,000 W	4,000 W	5,000 W
Max. output power	3,000 VA	4,000 VA	5,000 VA
Max. power from grid to battery	3,000 VA	4,000 VA	5,000 VA
Max. AC current ( $I_{ac,max}$ )	4.8 A	6.4 A	8.0 A
Grid connection (voltage range)		3-NPE 400 V / 230 V or 3-NPE 380 V / 220 V (+20 % / -30 %)	
Frequency (frequency range)		50 Hz / 60 Hz (45 - 65 Hz)	
Total harmonic distortion		< 3 %	
Power factor ( $\cos \varphi_{ac}$ )		0.85 - 1 incl. / cap.	

Obrázek A.1. Parametry střídače Fronius SYMO 5.0-3-S část 1 [18]

### TECHNICAL DATA FRONIUS SYMO HYBRID

GENERAL DATA	SYMO HYBRID 3.0-3-S	SYMO HYBRID 4.0-3-S	SYMO HYBRID 5.0-3-S
Dimensions (height x width x depth)	645 x 431 x 204 mm		
Weight	19.9 kg		
Degree of protection	IP 65		
Protection class	1		
Overvoltage category (DC / AC) <sup>1)</sup>	2 / 3		
Inverter design	Transformerless		
Cooling	Regulated air cooling		
Installation	Indoor and outdoor installation		
Ambient temperature range	-25 - +60°C		
Permitted humidity	0 - 100 %		
Max. altitude	2,000 m (unrestricted voltage range)		
DC PV connection technology	2x DC+ and 2x DC- screw terminals 2.5 - 16 mm <sup>2</sup>		
DC battery connection technology	1x DC+ and 1x DC- screw terminals 2.5 - 16 mm <sup>2</sup>		
AC connection technology	5-pin AC screw terminals 2.5 - 16 mm <sup>2</sup>		
Certificates and compliance with standards	VDE AR N 4105, ÖVE / ÖNORM E 8001-4-712, DIN V VDE 0126-1-1		
Stand alone	Yes		
Emergency power function switchover time	5 sec.		

EFFICIENCY	SYMO HYBRID 3.0-3-S	SYMO HYBRID 4.0-3-S	SYMO HYBRID 5.0-3-S
Max. efficiency (PV - grid)	97.5 %	97.6 %	
Max. efficiency (PV - battery - grid)	> 90.0 %	> 90.0 %	> 90.0 %
Europ. efficiency (PV - grid)	95.2 %	95.7 %	96.0 %

PROTECTION DEVICES	SYMO HYBRID 3.0-3-S	SYMO HYBRID 4.0-3-S	SYMO HYBRID 5.0-3-S
DC disconnect	Included		
Overload behaviour	Operating point shift, power limitation		
DC insulation measurement	Included		
Integral RCMU	Yes		

INTERFACES	SYMO HYBRID 3.0-3-S	SYMO HYBRID 4.0-3-S	SYMO HYBRID 5.0-3-S
WLAN / Ethernet LAN	Fronius Solar.web, Modbus TCP SunSpec, Fronius Solar API (JSON)		
Datalogger and web server	Included		
Interface to battery and meter	Modbus RTU SunSpec (RS485)		

<sup>1)</sup> Testing to IEC 62109-1.

Obrázek A.2. Parametry střídače Fronius SYMO 5.0-3-S část 2 [18]

### TECHNICAL DATA FRONIUS SMART METER

/ The Fronius Smart Meter is a bidirectional meter which optimises self-consumption and records the household's load curve. In conjunction with the Fronius Solar.web online portal, the Fronius Smart Meter provides a clear overview of a user's own power consumption.



GENERAL DATA	FRONIUS SMART METER
Nominal voltage	400 - 415 V
Max. current	3 x 63 A
Cable cross-section, power path	1 - 16 mm <sup>2</sup>
Cable cross-section, communication	0.05 - 4 mm <sup>2</sup>
Installation	DIN rail
Housing (width)	4 modules DIN 43880 (4 pole)
Accuracy class	1
Interface to inverter	Modbus RTU (RS485)
Display	8-digit LCD

Obrázek A.3. Parametry zařízení Fronius Smart Meter [18]

## TECHNICAL DATA FRONIUS SOLAR BATTERY

/ The Fronius Solar Battery is a perfect example of high-performance lithium iron phosphate technology. A long service life, short charging times and high depth of discharge are therefore guaranteed. The storage capacity of the Fronius Solar Battery can be adapted to meet individual customer needs.



ELECTRICAL PARAMETERS	BATTERY 4.5	BATTERY 6.0	BATTERY 7.5	BATTERY 9.0	BATTERY 10.5	BATTERY 12.0
Nominal capacity	4.5 kWh	6.0 kWh	7.5 kWh	9.0 kWh	10.5 kWh	12.0 kWh
Usable capacity (80% DoD)	3.6 kWh	4.8 kWh	6.0 kWh	7.2 kWh	8.4 kWh	9.6 kWh
Cycle stability (80% DoD)	8,000					
Voltage range	120 - 170 V	160 - 230 V	200 - 290 V	240 - 345 V	280 - 400 V	320 - 460 V
Nominal charging power	2,400 W	3,200 W	4,000 W	4,800 W	5,600 W	6,400 W
Nominal discharge power	2,400 W	3,200 W	4,000 W	4,800 W	5,600 W	6,400 W
Max. charging current	16 A					
Max. discharge current	16 A					
GENERAL DATA	BATTERY 4.5	BATTERY 6.0	BATTERY 7.5	BATTERY 9.0	BATTERY 10.5	BATTERY 12.0
Battery technology	LiFePO <sub>4</sub>					
Dimensions (height x width x depth)	955 x 570 x 611 mm					
Weight	91 kg	108 kg	125 kg	142 kg	159 kg	176 kg
Degree of protection	IP 20					
Protection class	1					
Installation type	Indoor installation					
Ambient temperature range	5 - 35°C					
Permitted humidity	0 - 100 %					
DC connection technology	Screw terminals 2.5 - 16 mm <sup>2</sup>					
Certificates and compliance with standards	IEC/EN 62133; EN 61000-6-2:2005, EN 61000-6-3:2007 + A1:2011, EN 62311:2008, FCC Part 15 Subpart B:2012 ClassB, UN 38.3					
INTERFACES	BATTERY 4.5	BATTERY 6.0	BATTERY 7.5	BATTERY 9.0	BATTERY 10.5	BATTERY 12.0
Connection to inverter	Modbus RTU SunSpec (RS485)					

Obrázek A.4. Parametry baterie Fronius [18]

# Příloha B

## Specifikace použitého solárního panelu

### SunForte PM096B00

Mono-Crystalline  
Photovoltaic Module

20%  
EFF.



320V  
333W

**Power Range**  
320 ~ 333 Wp



**Highly Strengthened Design**  
Module complies with advanced loading tests to meet 5400 Pa loading requirements



**Resistance to Salt Corrosion and Humidity**  
Module complies with IEC 61701: Salt Mist Corrosion Testing



**Back Contact Cells**  
No string in the front side enhances light conversion space



**IP-67 Rated Junction Box**  
Advanced water and dust proof level



**Transformer less**  
Validates the compatibility with transformer-less inverters at high system voltage.



**PID-Free**



**Superior Performance at High Temperatures**  
Less power loss in hot weather conditions due to the low temperature coefficient



ASSEMBLED IN  
TAIWAN



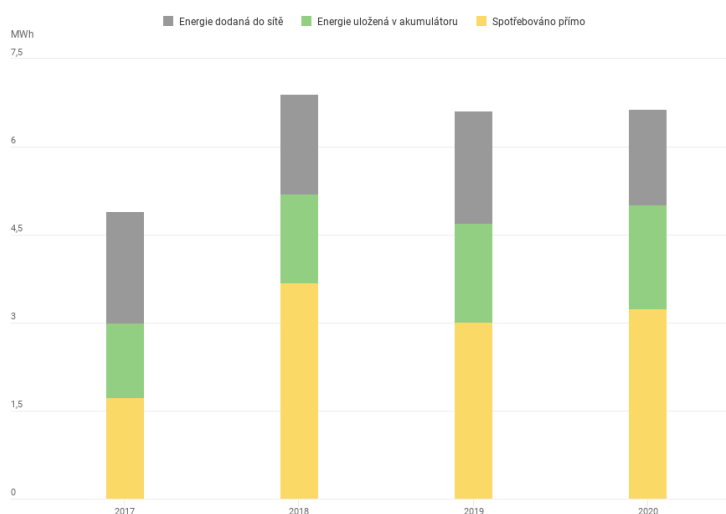
BenQ  
Solar



## Příloha C

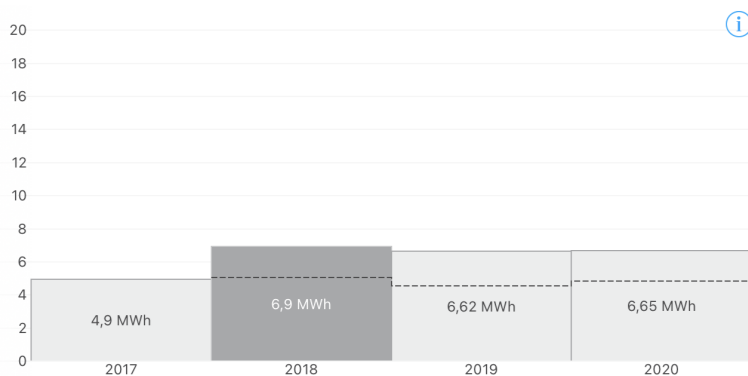
### Délka provozu fotovoltaické elektrárny

Fotovoltaická elektrárna byla v provozu od 21.04.2017 12:39 do 14.12.2020 09:13 a během této doby vyrobila 25,1 MWh elektrické energie, což odpovídá úspoře 13,3t  $CO_2$ , ujetých 88 566 km po dálnici nebo zasazení 341 stromů. Na obrázku C.5 je žlutě zobrazená energie spotřebovaná přímo, zeleně energie uložená v akumulátoru a šedě energie dodaná do vyřejné sítě.



**Obrázek C.5.** Přehled vyrobené elektrické energie a její umístění [3]

Při pohledu na přehled vyrobené elektrické energie dle let na obrázku C.6 a po nepřesném přepočtu<sup>1</sup> pro rok 2017 na 7,04 MWh a i při nedopočítávání chybějící části roku 2020 vykazuje množství vyrobené elektrické energie za jednotlivé roky podobné hodnoty. Vzhledem k termínu pro odevzdání diplomové práce i vzhledem k odstavení fotovoltaické elektrárny nebudou k dispozici další hodnoty pro přesnější výpočet.



**Obrázek C.6.** Přehled vyrobené elektrické energie dle let [3]

<sup>1</sup> V dopočítaných zimních měsících bývá výroba výrazně nižší než v létě.

Teď ještě zbývá vysvětlit, proč byla fotovoltaická elektrárna odstavena. Solární panely a jejich konstrukce byla instalovaná na betonové panely. Firma Solarinvest při instalaci několik panelů provrtala skrz. Po delším čase a v souvislosti se silnými poryvy větru na jaře roku 2020 mi začala do horního patra domu poprvé zatékat střechou voda. Reklamoval jsem dodávku fotovoltaické elektrárny a technik Solarinvestu zjistil, že jeden betonový panel byl provrtán skrz a šroub poškodil střechu. Situace se opakovala na konci léta 2020. Při druhé reklamaci znovu opravil původní poškození a našel další díru. Na konci roku 2020 se situace znovu opakovala. Vzhledem k tomu, že se jednalo o třetí stejnou závadu, jsem požádal demontování fotovoltaické elektrárny, opravu střechy a vrácení pořizovací ceny. Při kontrole střechy dorazili tři technici ze Solarinvestu a dva manažeři z PREměření. Na střeše bylo nalezeno celkem 11 děr od šroubů procházejícími betonovými panely. Solární panely byly demontovány, díry provizorně zalepeny a nyní řeší právní oddělení PRE, jak budou na moji reklamaci reagovat. Fotovoltaická elektrárna je mimo provoz. Můj řídicí program byl upraven na provoz bez střídače a je v provozu.



**Obrázek C.7.** Demontované solární panely při kontrole poškození střechy

Jsem rád za všechnu zelenou elektřinu, kterou jsem vyrobil. Věřím, že se zatékání vody do domu i finanční stránka reklamace vyřeší, a že se firma poučí z toho, co udělali špatně při instalaci na moji střechu. Za období skoro čtyř let provozu jsem od společnosti

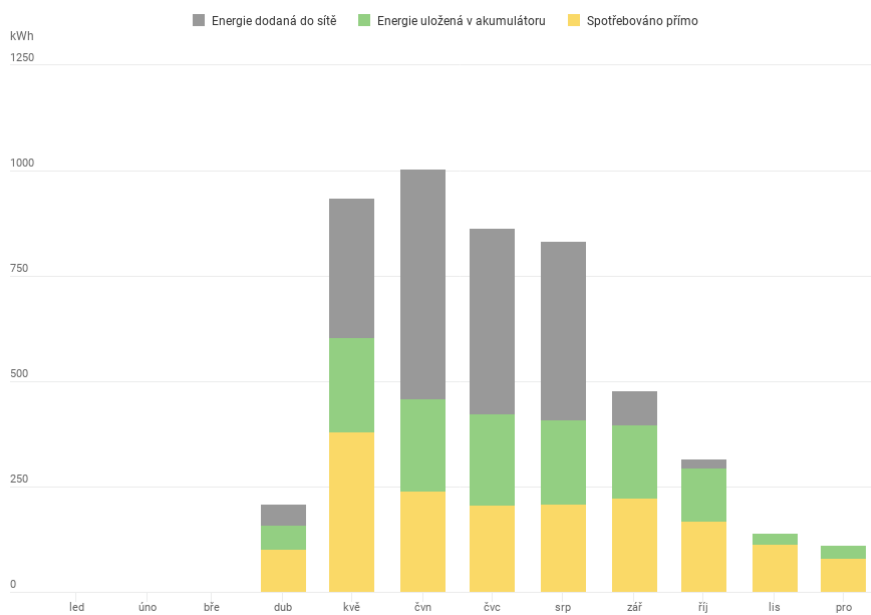
Fronius International GmbH obdržel certifikát za můj příspěvek k aktivitě 24 hodin slunce (anglicky 24hrs SUN). C.8)



**Obrázek C.8.** Certifikát od společnosti Fronius 24 hodin slunce [3]

# Příloha D

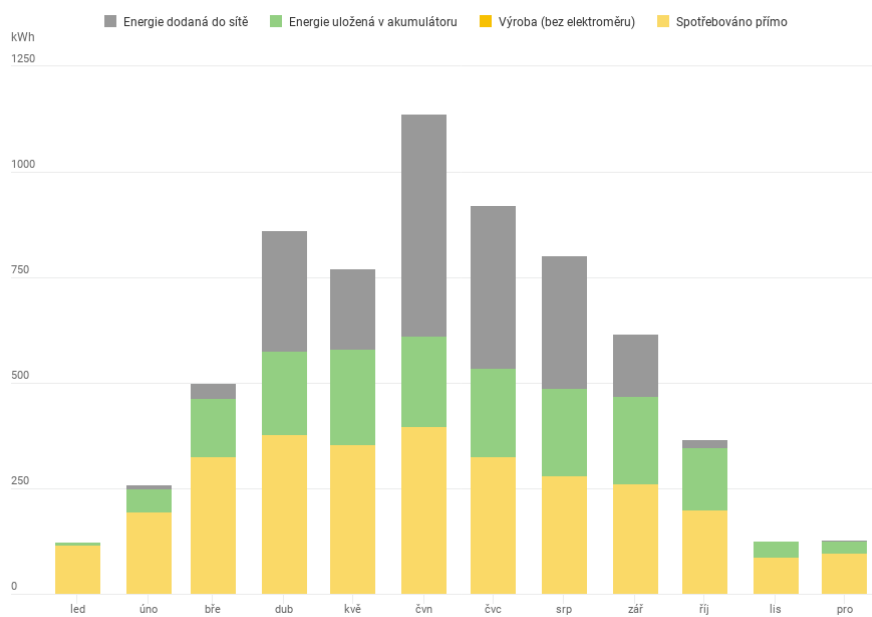
## Sezónnost výroby elektřiny ze slunce



Obrázek D.9. Spotřeba vyrobené elektřiny v roce 2017 dle následného využití



Obrázek D.10. Spotřeba vyrobené elektřiny v roce 2018 dle následného využití



**Obrázek D.11.** Spotřeba vyrobené elektřiny v roce 2019 dle následného využití



**Obrázek D.12.** Spotřeba vyrobené elektřiny v roce 2020 dle následného využití

# Příloha E

## Faktura za červenec 2020

► PŘEHLED PLATEB - vyúčtování záloh	
Přijetí platby	Platba vč. DPH (Kč)
10.7.2020	100,00
<b>Celkem</b>	<b>100,00</b>

\* Případné platby přijaté po datu odečtu 31.7.2020 budou automaticky vyúčtovány v dalším období.

PŘEHLED PLATEB dle vyhlášky č. 70/2016 Sb.			
Množství	Průměrná jednotková cena (bez DPH)**	Celkem bez DPH (Kč)	Celkem s DPH* (Kč)
VT 28 kWh	2,14 Kč/kWh	59,83	72,39
NT 142 kWh	2,06 Kč/kWh	293,14	354,70
1 měsíců	577,08 Kč/měsíc	577,08	698,27
<b>Celkem</b>		<b>930,05</b>	<b>1 125,36</b>

\* Háléřový rozdíl (proti částce v rekapitulaci) je způsoben zaokrouhlováním.  
\*\* Vypočítáno podílem částky celkem bez DPH a účtovaného množství.

### DETAIL SPOTŘEBY

Období		Měřidlo	Tarif	Stav měřidla		ZO*	TDD	Náso- bítel	Spotřeba (kWh)	Korekce (kWh)	Spotřeba v období	
od	do			počáteční	koncový						(kWh)	(MWh)
1.7.2020	31.7.2020	37768634	VT	2651	2679	D	7	1	28	0	28	0,02800
1.7.2020	31.7.2020	37768634	NT	45107	45249	D	7	1	142	0	142	0,14200
<b>Celkem dodané množství elektřiny</b>												<b>0,17000</b>

\* C - oprava spotřeby, D - odečet distributorem, O - odhad spotřeby distributorem, S - samoodečet, T - odhad spotřeby dle TDD, Z - změna ceny

### ► DETAIL REGULOVANÝCH PLATEB

Období		Položka	Sazba	Tarif	Množství	Jednotka	Za jednotku bez DPH	Celkem bez DPH (Kč)
od	do							
1.7.2020	31.7.2020	Distribuované množství elektřiny NT	D45d	NT	0,14200	MWh	164,93000 Kč	23,42
1.7.2020	31.7.2020	Distribuované množství elektřiny VT	D45d	VT	0,02800	MWh	237,63000 Kč	6,65
Období		Položka	Jistič (fáze x A)	Množství	Jednotka	Za jednotku bez DPH	Celkem bez DPH (Kč)	
od	do							
1.7.2020	31.7.2020	Měsíční plat za příkon - jistič	3x50	1,00000	měsíc	572,00000 Kč	572,00	
1.7.2020	31.7.2020	Cena za činnost zúčtování OTE		1,00000	měsíc	5,08000 Kč	5,08	
1.7.2020	31.7.2020	Cena na podporu výkupu elektřiny (OZE)		0,17000	MWh	495,00000 Kč	84,15	
1.7.2020	31.7.2020	Systémové služby ČEPS, a.s.		0,17000	MWh	77,12000 Kč	13,11	
1.7.2020	31.7.2020	Daň z elektřiny		0,17000	MWh	28,30000 Kč	4,81	
<b>Celkem (bez DPH)</b>								<b>709,22</b>

### ► DETAIL OBCHODNÍCH PLATEB

EL | D | START++ | PRAHA | HOME PRÍMOTOP 20 | D45d | 1.10.2019

Období		Položka	Sazba	Množství	Jednotka	Za jednotku bez DPH	Celkem bez DPH (Kč)	
od	do							
1.7.2020	31.7.2020	Silová elektřina, Pevná cena za měsíc	D45d	1,00000	měsíc	0,00000 Kč	0,00	
1.7.2020	31.7.2020	Silová elektřina, Spotřeba NT	D45d	0,14200	MWh	1 299,00000 Kč	184,46	
1.7.2020	31.7.2020	Silová elektřina, Spotřeba VT	D45d	0,02800	MWh	1 299,00000 Kč	36,37	
<b>Celkem (bez DPH)</b>								<b>220,83</b>

### OBCHODNÍ CENA ZA VÝKUP SILOVÉ ELEKTŘINY

EAN 859 182 400 3048 11857

Období*		Položka	Množství	Jednotka	Za jednotku bez DPH	Celkem (není předmětem daně) (Kč)
od	do					
1.7.2020	31.7.2020	Výkup elektřiny dodavatelem	0,47300	MWh	-839,98750 Kč	-397,31
Období*		Položka	Množství	Jednotka	Za jednotku bez DPH	Celkem bez DPH (Kč)
od	do					
1.7.2020	31.7.2020	Cena za službu výkupu elektřiny	0,47300	MWh	300,00000 Kč	141,90

\*Období výkupu je kráceno do posledního ukončeného měsíce fakturované dodávky. Chybějící část období bude zahrnutá v následujícím vyúčtování.

Obrázek E.13. Strana č. 2 z faktury od BOHEMIA ENERGY entity, s. r. o. za červenec 2020

# Příloha F

## Řízení letového provozu České republiky



Pan  
Jiří Navrátil  
Novosuchdolská 581/5a  
165 00 Praha 6 - Suchdol

Váš dopis zn. / dne:	Naše zn.:	Vyřizuje:	T: 220 372 002	Jeneč, datum:
Web/8.4.2019	SCL/4729/2019	Drhová K. Ing.	E: <a href="mailto:drhova@ans.cz">drhova@ans.cz</a>	11.04.2019
zaevidováno dne	souvisí s č. ž.		Mobil: 777 66 33 68	
09.04.2019	SCL/4729/2019			

**Vyjádření ŘLP ČR, s. p. k žádosti k PD pro stavební řízení na akci: „Fotovoltaická elektrárna na RD, parc. č. 585, k.ú. Suchdol.“**


K výše uvedené žádosti nemá ŘLP ČR, s. p. připomínky.

Stavbou dotčená lokalita je dle Vámi dodaných podkladů mimo oblast našich zájmů a nenacházejí se zde žádná podzemní ani pozemní zabezpečovací zařízení, která jsou ve správě ŘLP ČR, s. p.

Toto vyjádření má platnost jeden rok od data svého vydání.

S pozdravem

Ing. Jan Sedlák  
Náměstek sekce centrální logistiky

  
Řízení letového provozu  
České republiky, státní podnik  
Navigační 787  
252 61 Jeneč, Česká republika  
IČO: 49710371, DIČ: CZ49710371  
DIČ skupinové pro účely ÚPH: CZ02004742

(10)

1/1

Řízení letového provozu  
České republiky, státní podnik  
Navigační 787, 252 61 Jeneč  
Česká republika  
E [info@rlp.cz](mailto:info@rlp.cz) / [www.rlp.cz](http://www.rlp.cz)

IČO: 49710371, DIČ: CZ49710371  
Registrováno Městským soudem v Praze  
odtíl A, vložka 10771  
Bankovní spojení: ČSOB Praha 5, č.ú. 88153/0300  
Držitel certifikátu kvality ISO 9001:2008



# Příloha G

## PostgreSQL schema.sql

```
-- výroba z FVE

CREATE TABLE IF NOT EXISTS vyroba_max (
    id SERIAL PRIMARY KEY,
    mtime TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP NOT NULL,
    hodnota SMALLINT DEFAULT 0
);
COMMENT ON TABLE vyroba_max IS 'denní maximální výroba z FVE';

-- rozsah výroby z FVE

CREATE TABLE IF NOT EXISTS vyroba_rozsah (
    id SERIAL PRIMARY KEY,
    datum DATE DEFAULT CURRENT_DATE NOT NULL,
    cas_od TIME NOT NULL,
    cas_do TIME NOT NULL
);
COMMENT ON TABLE vyroba_rozsah IS 'denní časový rozsah výroby z FVE';

-- odběr ze sítě

CREATE TABLE IF NOT EXISTS ze_site_min (
    id SERIAL PRIMARY KEY,
    mtime TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP NOT NULL,
    hodnota SMALLINT DEFAULT 32767
    -- 32767 je max. hodnota ve SMALLINT
);
COMMENT ON TABLE ze_site_min IS 'denní minima odběru (nákupu) ze sítě';

-- dodávky do sítě

CREATE TABLE IF NOT EXISTS do_site_max (
    id SERIAL PRIMARY KEY,
    mtime TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP NOT NULL,
    hodnota SMALLINT DEFAULT 0
    -- doposud známé max je 21514W tedy <= 32767 max SMALLINT
);
COMMENT ON TABLE do_site_max IS 'denní maxima dodávky do sítě';

-- baterie

CREATE TABLE IF NOT EXISTS baterie_min (
    id SERIAL PRIMARY KEY,
    mtime TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP NOT NULL,
```



```

        hodnota SMALLINT DEFAULT 10000
        -- 10000 je více, než maximální možná hodnota 7200
    );
    COMMENT ON TABLE baterie_min IS 'denní minima nabití baterie';

    CREATE TABLE IF NOT EXISTS baterie_max (
        id SERIAL PRIMARY KEY,
        mtime TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP NOT NULL,
        hodnota SMALLINT DEFAULT 0
    );
    COMMENT ON TABLE baterie_max IS 'denní maxima nabití baterie';

    CREATE TABLE IF NOT EXISTS stridac (
        id SERIAL PRIMARY KEY,
        mtime TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP NOT NULL,
        power_flow_realtime_data JSONB NOT NULL
        -- GetPowerFlowRealtimeData
    );
    COMMENT ON TABLE stridac IS 'JSON pro střídač';

    GRANT SELECT ON ALL TABLES IN SCHEMA public TO script;
    GRANT INSERT ON ALL TABLES IN SCHEMA public TO script;
    GRANT UPDATE ON ALL TABLES IN SCHEMA public TO script;

    GRANT USAGE ON ALL SEQUENCES IN SCHEMA public TO script;

    CREATE VIEW view_status AS
    SELECT
        timezone('CEST', now()) AS "Prague",
        (SELECT hodnota FROM vyroba_max WHERE DATE(mtime) = DATE(now()))
        AS vyroba_max_W,
        (SELECT hodnota FROM baterie_min WHERE DATE(mtime) = DATE(now()))
        AS min_bat_Wh,
        (SELECT hodnota FROM baterie_max WHERE DATE(mtime) = DATE(now()))
        AS max_bat_Wh,
        (SELECT hodnota FROM ze_site_min WHERE DATE(mtime) = DATE(now()))
        AS ze_site_min_W,
        (SELECT hodnota FROM do_site_max WHERE DATE(mtime) = DATE(now()))
        AS do_site_max_W
    ;

    GRANT SELECT ON view_status TO script;

```

# Příloha H

## Fronius webová a mobilní aplikace

Střídač posílá data o provozu průběžně na servery společnosti Fronius International GmbH. Některá data je možné následně prohlížet na webu <https://www.solarweb.com/> a v mobilní aplikaci Solar.web H.14. Grafy použité v této práci byly převzaty z těchto zdrojů.

O aplikaci Solar.web

Hotovo

---

### Imprint / disclosure

The Smartphone application and all information contained therein is provided by Fronius International GmbH.

**Main aim of the application:**

visualization of photo-voltaic-systems.

**Media owned and published by:**

Fronius International GmbH  
Vorchdorfer Straße 40  
A-4643 Pettenbach

**Nature and purpose of the media owner:**

development, production and sale of products in the battery charging, welding technology and solar electronics sectors.

**Registered office of the media owner:**

Vorchdorfer Straße 40  
A-4643 Pettenbach

**Contact details:**

E-Mail: [contact@fronius.com](mailto:contact@fronius.com)

Tel.: +43/7242/241-0

Fax: +43/7242/241-2670

**Registered number and registration office:**

Registered number: 149888z

Registration office: Steyr regional court

UID: ATU52614407

**Obrázek H.14.** Mobilní iOS aplikace Solar.web

## Příloha I

### Maximální výroba v roce 2020

```
psql -h 192.168.4.52 -U navratil -d rh
SSL connection (protocol: TLSv1.3, cipher: AEAD-AES256-GCM-SHA384,
bits: 256, compression: off)
rh=# SELECT hodnota AS "výroba kWp", mtime AS "datum a čas"
FROM vyroba_max ORDER BY hodnota DESC LIMIT 88;
```

výroba kWp	datum a čas
8045	2020-06-06 12:27:18.140169+02
7873	2020-05-26 12:33:20.376929+02
7860	2020-05-02 13:49:22.312084+02
7839	2020-05-25 12:41:15.724667+02
7724	2020-07-09 12:43:57.395054+02
7681	2020-05-22 12:15:41.971879+02
7566	2020-07-06 12:26:25.651268+02
7560	2020-05-06 12:04:49.405028+02
7491	2020-07-18 11:00:21.628374+02
7479	2020-05-28 14:07:26.104898+02
7444	2020-04-25 11:48:33.336468+02
7313	2020-08-27 11:52:33.657432+02
7284	2020-05-03 10:54:44.965609+02
7181	2020-06-05 12:52:57.727941+02
7143	2020-07-17 11:40:31.971685+02
7138	2020-06-25 12:35:35.554351+02
7108	2020-05-14 12:50:18.921311+02
7078	2020-04-14 12:03:28.277423+02
7078	2020-07-03 13:38:57.753374+02
6990	2020-05-23 12:59:50.163252+02
6962	2020-07-02 13:38:59.481109+02
6927	2020-05-01 11:34:40.975597+02
6924	2020-08-31 12:01:13.986268+02
6899	2020-07-11 14:42:32.503772+02
6875	2020-06-02 12:33:03.795302+02
6853	2020-07-16 11:40:00.548151+02
6825	2020-07-05 12:43:16.00803+02
6807	2020-05-20 13:56:43.945161+02
6719	2020-03-22 12:02:31.421356+01
6700	2020-03-31 12:25:10.935618+02
6680	2020-05-12 12:40:32.044582+02
6613	2020-05-24 10:38:01.893705+02
6573	2020-05-05 12:09:15.205558+02
6563	2020-07-21 13:48:22.028104+02
6563	2020-04-30 13:21:35.967123+02
6467	2020-05-27 12:12:56.712661+02

6454		2020-09-04	12:46:09.812158+02
6447		2020-06-28	12:52:56.545112+02
6442		2020-07-10	12:06:36.236508+02
6424		2020-06-01	13:00:28.507273+02
6404		2020-08-23	13:46:14.886306+02
6396		2020-05-04	14:24:53.254319+02
6367		2020-07-27	12:18:03.96256+02
6360		2020-08-28	11:14:07.387256+02
6312		2020-06-11	14:46:27.107285+02
6268		2020-04-19	13:21:07.402476+02
6256		2020-06-15	12:37:02.10296+02
6238		2020-05-31	15:14:56.473675+02
6173		2020-07-22	13:13:11.808619+02
6172		2020-07-28	12:48:12.495514+02
6168		2020-06-17	10:33:30.774456+02
6162		2020-08-25	13:46:45.663931+02
6152		2020-05-17	12:16:41.275756+02
6150		2020-07-13	11:39:07.347137+02
6145		2020-08-18	12:17:00.874189+02
6116		2020-07-23	13:11:09.37683+02
6111		2020-06-07	13:00:04.586097+02
6096		2020-03-12	12:10:32.011404+01
6060		2020-06-22	12:41:48.598059+02
6042		2020-05-16	11:23:41.05346+02
6035		2020-05-21	12:15:29.62671+02
6010		2020-06-24	11:33:56.208414+02
5979		2020-04-28	13:00:55.679431+02
5934		2020-06-23	13:18:29.959402+02
5918		2020-09-11	11:41:47.968527+02
5916		2020-04-10	12:26:53.254125+02
5887		2020-04-13	12:15:33.675508+02
5876		2020-07-24	10:43:49.794851+02
5840		2020-06-04	11:35:29.153192+02
5833		2020-09-02	12:26:16.316515+02
5832		2020-03-26	12:07:48.776187+01
5806		2020-06-08	10:28:35.201774+02
5796		2020-07-12	11:19:55.237416+02
5783		2020-09-07	13:35:37.808728+02
5757		2020-07-01	11:40:46.682365+02
5748		2020-06-03	13:27:41.343648+02
5744		2020-08-24	12:10:24.52286+02
5700		2020-03-29	12:44:40.21244+02
5697		2020-09-03	13:14:37.854356+02
5675		2020-07-04	13:37:49.961725+02
5655		2020-10-06	13:16:56.333638+02
5654		2020-02-28	11:12:02.098571+01
5640		2020-06-27	13:56:38.643267+02
5623		2020-05-13	12:41:00.852149+02
5622		2020-07-29	11:10:28.432971+02
5620		2020-03-13	11:34:06.081808+01
5586		2020-05-08	12:33:25.361621+02
5586		2020-07-08	10:27:01.344214+02

(88 rows)

# Příloha J

## Porovnání ceny FVE v letech 2016 a 2021

V této práci použitou FVE jsem objednával v roce 2016 a celkově jsem za ní k 25.09.2017 zaplatil 570 620 Kč. Od SFŽP ČR jsem k 16.11.2017 obdržel dotaci 155 000 Kč. Pokud by se FVE s obdobnými parametry objednávala nyní, tj. v únoru 2021, tak by stála 387 424 Kč. Výše dotace zůstala stejná.

NOVÁ ZELENÁ ÚSPORÁM		INDIVIDUÁLNÍ NABÍDKA SOLÁRNÍHO FOTOVOLTAICKÉHO SYSTÉMU		SOLAR INVEST SOLÁRNÍ SYSTÉMY NA KLÍČ		PRE SKUPINA		
NABÍDKA PRO:		Vážený pan Jiří Navrátil		VYPRACOVAL:				
Adresa realizace:		Praha		Datum nabídky: ČÍSLO :		26.02.2021 2021/02		
				Nabídka platí do:		12.03.2021		
Dotace NZU ( C.3.7) - XL aku		dotace ve výši: 150 000 Kč		VELIKOST SYSTÉMU kWp 5,04				
výsledná cena investice 237 424,- Kč		Region (SM/SC) NE		Meziroční výroba elektřiny kWh/rok 5040				
<b>Nabídka fotovoltaického systému na klíč:</b>								
						počet ks	cena celkem bez DPH	
<b>Na fotovoltaický systém použijeme následující fotovoltaické panely</b>								
AEG AS-M 1203B-H-360 / celočerný, half-cut (1765x1048x35 mm)SVT25419						360 Wp	14	51 530 Kč
<b>Fotovoltaické (hybridní střídače)</b>								
GoodWe ET series / Hybrid - 3f/asym/ 8kW / 3F / HYB						49 752 Kč	1	49 752 Kč
SolarEdge SESK-N6 (5kW) / 3f / Optimizery							0	- Kč
Optimezer P700- 5RM4MRX / / Optimizery							0	- Kč
<b>Akumulátory a příslušenství</b>								
Battery set Pylontech Force H2 - BMS+2 (7,1 kWh)BAT / LiFePO4						91 161 Kč	1	91 161 Kč
						- Kč	0	- Kč
<b>Ostatní komponenty</b>								
Montážní systém (konstrukce) pro šikmé střechy -special						1 999,00 Kč	14	27 986 Kč
Watt router Mx/regulator přebyteků, 3xSSR 20A, montáž a spotř. Mat. (pro 3f FVE)						11 999,00 Kč	0	- Kč
elektro materiál, rozvaděč, kabeláže a ostatní elektro (ostatní hybrid do 10kW)						26 500,00 Kč	1	26 500 Kč
stacionární zásobník 200ltr + 3f patrona 3-6kW,						13 590,00 Kč	0	- Kč
přepětové ochrany + DC rozvaděč (sada pro 2MPP)						7 990,00 Kč	1	7 990 Kč
<b>Práce a další služby</b>								
Projektová dokumentace, revize, zpráva, ostatní zkoušky						5 000,00 Kč	1	5 000 Kč
Zastupování (PDS / distribuční spol., Licence ERU (pokud je třeba), NZU,.... (neobsahuje energetický posudek pro NZU, kolký a případné poplatky)						1,00 Kč	1	1 Kč
<b>Montážní práce</b> (instalace panelů), elektroinstalace, technická kontrola a uvedení do provozu, zkušební provoz a nastavení syst.,monitoringu, regulace						44 380,00 Kč	1	44 380 Kč
Instalace nádrže na TUV (napojení na systém TUV) - plastové vedení standard						4 000,00 Kč	0	- Kč
Doprava osob, logistika materiálu a technologií (komplet) Hybrid s AKU						9 990,00 Kč	1	9 990 Kč
příplatek za kabelovou smyčku (pro Backup) / cena za metr nad standard (standard 12 bm)						381,00 Kč	0	- Kč
Montážní systém (konstrukce) pro rovné střechy -PMT (Jih 10 a 15st.)						2 638,00 Kč	0	- Kč
Úprava odběrného místa (včetně revize)						11 900,00 Kč	1	11 900 Kč
Lešení						10 000,00 Kč	1	10 000 Kč
Recyklační poplatek dle zákona o odpadech / kolektivní recyklační systém						50,00 Kč	14	700 Kč
<b>Cena základní sestavy bez DPH</b>						<b>336 890,- Kč</b>		
<b>DPH 15%</b>						<b>50 534,- Kč</b>		
<b>Cena celkem</b>						<b>387 424,- Kč</b>		
<b>Cena celkem po odečtu dotace</b>						<b>237 424,- Kč</b>		

v002.00/2020\_11

## Příloha K

### Rezervace elektromobilu Tesla Model 3

Elon Musk představil dne 31. března 2016 prototyp elektromobilu Tesla Model 3. Cena pro uvedení na americký trh byla stanovena na 35 000 USD. Zlákala mne představa, že budu vlastnit elektromobil a nabíjet ho z fotovoltaické elektrárny. Dne 12. května 2016 jsem se nechal se zapsat do pořadníku a uhradil 1 000 €. Očekávaný termín dodání se postupně oddaloval, což mi nevadilo. Nakonec mi dne 3. října 2019 v 09:09 přišel email z buildmy3EMEA@tesla.com s předmětem „Navrhněte si svůj Model 3“ a obsahem „Model 3 lze nyní objednat v České republice. Je čas vybrat si svůj vůz a zadat objednávku pro dodání počátkem roku 2020. Tesla, Freyova 980/10, 190 00 Praha 9“. Nejlevnější konfigurace pro český trh bohužel začínala na 1 199 200 Kč, a tak jsem 4. října 2019 požádal o zrušení rezervace a 29.10.2019 jsem obdržel 1 000 € zpět.



#### Your Model 3 Delivery Estimate

Model 3 deliveries are beginning in the US and Canada based on the time and date that you placed your reservation.

RN10805    5 made on May 12, 2016

**Early 2019**

**Obrázek K.15.** Rezervace elektromobilu Tesla Model 3

## Příloha L

# Situace s fotovoltaikou v Praze

Tato příloha přebírá informace z článku ČTK [34] z 23.03.2021<sup>1</sup>

V Praze zhruba 600 000 domácností a na střechách je v provozu asi 1 600 fotovoltaických elektráren. Jen rodinné a bytové domy (tedy bez kancelářských, průmyslových a veřejných budov) mohou podle článku vyprodukovat přes 482 gigawatthodin (GWh) elektrické energie. Analýza brala v potaz i omezení, jako je orientace střechy, zastínění nebo neochota vlastníků bytových jednotek ke společné investici. Podle dat Energetického regulačního úřadu dodaly v roce 2019 všechny solární elektrárny v Praze v do sítě 21,9 GWh<sup>2</sup>

Analýza je součástí Manuálu pro využití lokální energetiky, který zmiňuje příklady ze zahraničních metropolí. Například ve Vídni budou masivně rozšiřovat solární panely na celkem 120.000 metrů čtverečních. Berlín má pak v plánu zajistit značné rozšíření solárních panelů na střechách města s cílem dodat do roku 2050 solární energii, která pokryje přibližně čtvrtinu potřeby města.

Instalovaný výkon solárních elektráren v ČR od roku 2012 stagnuje nebo roste jen mírně. Podle odborníků je důvodem ukončení podpory oboru obnovitelných zdrojů po takzvaném solárním boomu. V posledních měsících navzdory dopadům šíření koronaviru však roste zájem právě o fotovoltaické instalace na střechy domů a podniků, shodli se už dříve největší prodejci na trhu.

<sup>1</sup> Článek vyšel pár dní před odevzdáním této práce.

<sup>2</sup> Pro porovnání moje FVE v roce 2019 dodala do sítě 6,62 MWh.