

Mikropočítačová jednotka pro optimalizaci toku energie v domácnosti s fotovoltaickou elektrárnou

Jan Kopecký

Bakalářská práce
2025



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav automatizace a řídicí techniky

Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Jan Kopecký
Osobní číslo:	A22173
Studijní program:	B0714A150006 Aplikovaná informatika v průmyslové automatizaci
Specializace:	Průmyslová automatizace
Forma studia:	Prezenční
Téma práce:	Mikropočítačová jednotka pro optimalizaci toku energie v domácnosti s fotovoltaickou elektrárnou
Téma práce anglicky:	Microcomputer Unit for Optimizing Energy Flow in a Household with a Photovoltaic Power Plant

Zásady pro vypracování

- Zpracujte literární rešerši na dané téma.
- Proveďte analýzu současného stavu monitoringu a řízení fotovoltaické elektrárny.
- Navrhněte způsob implementace mikropočítačové jednotky pro optimalizaci toku energie do stávajícího systému.
- Vytvořte obslužné programové vybavení umožňující na základě aktuálních spotových cen elektrické energie a predikce osvitu řídit směr toku energie.
- Zrealizujte webovou vizualizaci pro monitorování provozu systému.
- Ověřte správnou funkci jednotky.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. BRADT, Paul a BRADT, David. *Science and engineering projects using the Arduino and Raspberry Pi: explore STEM concepts with microcomputers*. [United States]: Apress, 2020. ISBN 9781484258118.
2. HENZE, Andreas a Werner HILLEBRAND. *Elektrický proud ze slunce: fotovoltaika v praxi : technika, přehled trhu, návody ke stavbě*. Ostrava: HEL, 2000, 136 s. ISBN 8086167127.
3. MANN, Burkhard. *Č pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy – linkery, práce s ATME1 AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky*. Praha: BEN – technická literatura, 2003. μ C & praxe. ISBN 80-7300-077-6.
4. Orbit Merret. *OMC 8000: Základní modul PLC systému*. Online. 2016 [cit. 2024-10-21], s. 44. Dostupné z: https://www.orbitmerret.eu/en/document-download?document_id=5006.
5. Solax Power Network Technology. *Uživatelská příručka GBC Solino řady X3-Hybrid*. Online. 2022 [cit. 2024-10-21], s. 96. Dostupné také z: https://gbc-solino.cz/wp-content/uploads/2022/11/MNL_SOLAX_X3_HYBRID_G4_5.0-15.0-CZ-2.pdf.
6. Solax Power Network Technology. *Energy Storage Inverter Modbus TCP&RTU Communication protocols*. Online. 2021 [cit. 2024-10-21], s. 52. Dostupné také z: https://gbc-solino.cz/wp-content/uploads/2022/07/Hybrid-X1X3-G4-ModbusTCPRTU-V3.21-English_0622-public-version.pdf.

Vedoucí bakalářské práce: **Ing. Petr Dostálek, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **13. prosince 2024**

Termín odevzdání bakalářské práce: **27. května 2025**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 13. prosince 2024

PROHLÁŠENÍ AUTORA BAKALÁŘSKÉ PRÁCE

Beru na vědomí, že

- odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení bakalářské práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – bakalářskou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že

- jsem na bakalářské práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....
podpis autora

ABSTRAKT

Tato bakalářská práce se zaměřuje na návrh a realizaci systému pro efektivní řízení toku energie v domácnosti s fotovoltaickou elektrárnou. Cílem je implementovat do stávajícího systému řízení mikropočítač Raspberry Pi a vyvinout program, který optimalizuje využití přebytečné vyrobené energie z fotovoltaické elektrárny. Optimalizace spočívá v prodeji přebytečné eklektické energie za co nejvýhodnější spotové ceny daného dne a zákaz přetoků energie do eklektické sítě v době, kdy jsou spotové ceny záporné. Toho je docíleno komunikací s elektrickým střídačem pomocí protokolu Modbus, pro zjišťování aktuálních a denních statistik systému. Spotové ceny a predikce slunečního osvětlení jsou získávány z webu. Součástí práce je také vizualizace, která slouží pro zobrazení výsledků řízení a jejich kontrolu.

Klíčová slova: PLC, mikropočítač, spotové ceny, predikce slunečního osvětlení, vizualizace

ABSTRACT

This bachelor's thesis focuses on the design and implementation of a system for efficient energy flow management in a household equipped with a photovoltaic power plant. The goal is to integrate a Raspberry Pi microcomputer into the existing control system and develop a program that optimizes the use of surplus energy generated by the photovoltaic system. The optimization consists of selling the surplus electrical energy at the most favorable spot prices of the day and preventing energy feed-in to the electrical grid during periods when spot prices are negative. This is achieved through communication with the inverter using the Modbus protocol to retrieve current and daily system statistics. Spot prices and solar irradiation forecasts are retrieved from the web. The thesis also includes a visualization component, which serves to display and monitor the results of the control system.

Keywords: PLC, microcomputer, spot prices, solar irradiation forecast, visualization

Rád bych poděkoval především Ing. Petrovi Dostálkovi, Ph.D., vedoucímu této bakalářské práce, za odborné vedení, cenné připomínky a vstřícný přístup. Zvláštní poděkování patří také mému otci, Tomáši Kopeckému, jenž se významně podílel na vytvoření samotného zadání této bakalářské práce a současně poskytl finanční podporu nezbytnou pro její realizaci. V neposlední řadě děkuji své rodině a přátelům za trpělivost a podporu během celého studia.

OBSAH

ÚVOD.....	11
I TEORETICKÁ ČÁST	12
1 ENERGETIKA.....	13
1.1 TEPELNÁ ELEKTRÁRNA	13
1.2 JADERNÁ ELEKTRÁRNA	15
1.3 ELEKTRÁRNY NA BIOMASU	17
1.4 VODNÍ ELEKTRÁRNY	18
1.4.1 Průtočné vodní elektrárny	19
1.4.2 Akumulační vodní elektrárny.....	20
1.4.3 Přecherňávající vodní elektrárny.....	20
1.5 VĚTRNÉ ELEKTRÁRNY	20
1.6 FOTOVOLTAICKÉ ELEKTRÁRNY	21
1.6.1 Domáci FVE.....	22
1.6.2 Dostupná řízení fotovoltaické elektrárny	24
2 HARDWARE.....	25
2.1 MIKROPOČÍTAČ	25
2.1.1 Raspberry Pi	26
2.1.2 Raspberry Pi 5	27
2.1.3 Vývojové prostředí Visual Studio Code	28
2.2 PLC.....	29
2.2.1 Dělení PLC	30
2.2.2 Vývojové prostředí Multiprog.....	31
2.3 SBĚRNICE RS485	32
2.3.1 Modbus.....	32
II PRAKTICKÁ ČÁST	34
3 ANALÝZA SOUČASNÉHO STAVU SYSTÉMU	35
3.1 SOUČASNÉ ŘÍZENÍ SYSTÉMU	36
3.2 OPTIMALIZACE ŘÍZENÍ SYSTÉMU	40
4 POPIS REALIZACE	43
4.1 VÝBĚR MIKROPOČÍTAČE.....	43
4.2 INSTALACE MIKROPOČÍTAČE	43
4.3 OPERAČNÍ SYSTÉM MIKROPOČÍTAČE A VZDÁLENÝ PŘÍSTUP.....	44
4.3.1 Připojení VS Code pro vzdálený přístup.....	45
4.4 VÝVOJ PROGRAMU MIKROPOČÍTAČE	45
4.4.1 Stahování spotových cen.....	45
4.4.2 Stahování predikce osvitu	46

4.4.3	Komunikace mikropočítače se střídačem	47
4.4.4	Komunikace mikropočítače s PLC	49
4.4.5	Databáze	50
4.4.6	Hlavní program.....	51
4.5	ÚPRAVA PROGRAMU PLC.....	55
4.6	VIZUALIZACE	56
5	OVĚŘENÍ FUNKČNOSTI JEDNOTKY	60
	ZÁVĚR.....	63
	SEZNAM POUŽITÉ LITERATURY	65
	SEZNAM OBRÁZKŮ	70
	SEZNAM TABULEK.....	71
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	72
	SEZNAM PŘÍLOH.....	74

ÚVOD

V posledních letech roste zájem o efektivní využití obnovitelných zdrojů energie, což je do značné míry důsledkem rostoucích cen elektřiny, snahy o energetickou soběstačnost domácností a celosvětového důrazu na snižování emisí skleníkových plynů. Fotovoltaické elektrárny (FVE) se tak stávají běžnou součástí rodinných domů. Samotná výroba elektrické energie však k dosažení skutečné soběstačnosti nestačí. Klíčovou roli hraje také inteligentní řízení a optimalizace toku energie, které umožňuje maximálně využít vlastní výrobu a minimalizovat náklady spojené s odběrem z distribuční sítě.

Tato bakalářská práce se zaměřuje na návrh a realizaci systému pro řízení toku elektrické energie v domácnosti vybavené fotovoltaickou elektrárnou a akumulací energie. Cílem je nejen efektivní využití přebytečné vyrobené elektřiny, ale také optimalizace jejího prodeje v závislosti na aktuálních spotových cenách elektřiny na trhu. V obdobích, kdy jsou spotové ceny záporné, systém dokáže řízeně omezit přetoky do sítě. Významnou součástí řešení je predikce slunečního osvětlení, která umožňuje lépe plánovat využití akumulace a spotřeby v rámci jednotlivých dnů.

Pro lepší pochopení problematiky je práce rozdělena do několika částí. V teoretické části je nejprve popsána oblast energetiky, přičemž je věnována pozornost jednotlivým typům elektráren a podrobněji pak fotovoltaickým elektrárnám a jejich využití v domácnostech. Dále je uveden přehled použitého hardwaru, zejména mikropočítače Raspberry Pi, programovatelného logického automatu (PLC) a komunikace pomocí sběrnice RS485 a protokolu Modbus.

Praktická část se zaměřuje na analýzu současného stavu řízení toku energie v domácnosti, identifikaci nedostatků a návrh optimalizace systému. Následuje popis realizace, zahrnující instalaci mikropočítače, vývoj programového vybavení pro komunikaci se střídačem a PLC, stahování spotových cen a predikce slunečního osvětlení. Součástí je také vytvoření databáze pro ukládání dat a návrh vizualizace pro zpětnou kontrolu systému. Na závěr jsou shrnuty dosažené výsledky a zhodnocena funkčnost navrženého řešení.

I TEORETICKÁ ČÁST

1 ENERGETIKA

Energetika je odvětví, které se zabývá získáváním, přeměnou a distribucí všech forem energie. To znamená, že se stará o těžbu a distribuci uhlí, ropy, zemního plynu a dalších paliv, což navazuje na jejich zpracování, tedy výrobu a distribuci elektřiny, tepla a pohonných hmot.

V dnešní době je velmi významným tématem, hlavně v Evropské unii, emise skleníkových plynů. Právě energetika, tedy těžba a spalování fosilních paliv stojí za $\frac{3}{4}$ celosvětových emisí skleníkových plynů, zejména oxidu uhličitého (CO_2). Proto se nejen Česká republika snaží o náhradu hlavně tepelných elektráren za elektrárny z obnovitelných zdrojů či jaderné elektrárny. [1]

V České republice máme dva majoritní způsoby výroby elektrické energie, a to z hnědého uhlí a jádra. Hnědé uhlí tvoří přibližně 44 % výroby a jádro 37 %. Mezi ty minoritní pak patří plyn, 6 %, a obnovitelné zdroje, 14 %, které tvoří výroba elektřiny z biomasy a bioplynu, a také vodní, větrné a fotovoltaické elektrárny. [2]

1.1 Tepelná elektrárna

Na počátku rozvoje tepelných elektráren stál v roce 1884 vynález parní turbíny, který vytvořil irský inženýr Charles Algernon Parsons. Tato turbína byla zpočátku využívána jako součást pohonných systémů lodí, avšak brzy si našla uplatnění i v elektrárnách, kde nahradila méně účinné parní stroje. Jejím hlavním principem je přeměna tepelné energie vodní páry na mechanickou energii, přičemž rotační pohyb turbíny slouží k výrobě elektrické energie prostřednictvím připojeného alternátoru. [3]

Základním zařízením každé tepelné elektrárny je spalovací kotel, ve kterém dochází ke spalování paliva za účelem produkce tepla. Toto teplo je v první fázi přenášeno do uzavřeného vodního okruhu, kde se voda přeměňuje na páru. Pára pod vysokým tlakem následně proudí na lopatky turbíny, kde dochází k přeměně tepelné energie na mechanickou ve formě otáčivého pohybu turbíny, která je přímo spojena s generátorem vyrábějícím elektrickou energii. Po vykonání práce pára vstupuje do kondenzátoru, kde se opět mění na kapalinu, a celý proces se opakuje. [3]

K samotné výrobě elektrické energie neboli elektrickému proudu dochází díky Faradayovu indukčnímu zákonu. Tento jev vysvětluje princip elektromagnetické indukce, při kterém se ve vodiči, pohybujícím se vůči magnetickému poli, vytváří elektrické napětí.

Pokud je obvod uzavřený, začne vodičem procházet elektrický proud. Relativní pohyb může nastat nejen tehdy, když se pohybuje vodič, ale i v případě, že magnetické pole se pohybuje vůči vodiči. [3]

Mezi materiály, které může tepelná elektrárna spalovat, jsou nejčastější lignit, hnědé uhlí, černé uhlí, biomasa, ropné produkty a zemní plyn. Způsob spalování závisí na konstrukci kotle. Starší typy kotlů (roštové nebo práškové) mají menší účinnost a uhlí se spaluje při teplotách 1400-1600 °C. U této technologie dochází k velké emisi jak oxidu uhličitého, tak také oxidů síry. Proto patří k největším znečišťovatelům z hlediska ochrany životního prostředí. Modernější technologie, s vyšší účinností, je spalování ve fluidních kotlích, kde je teplota okolo 850 °C. Během spalovacího procesu se kromě paliva přidává také mletý vápenec, který slouží k odstranění oxidů síry. Tento proces vede ke vzniku tzv. energosádrovce, jenž nachází uplatnění zejména ve stavebnictví. Současně dochází ke snížení množství oxidů dusíku a dalších škodlivých plynů, což přispívá ke snižování ekologické zátěže. [3]

V roce 2020 dosáhla výroba elektrické energie v České republice 81,4 GWh, přičemž přibližně 55 % tohoto množství připadlo na tepelné elektrárny. Tento podíl je stále poměrně vysoký, přičemž většina těchto elektráren spaluje uhlí. To znamená, že z pohledu emisí oxidu uhličitého je česká energetika méně šetrná k životnímu prostředí. Je však třeba dodat, že změna energetického mixu je proces vyžadující značné finanční prostředky a dlouhodobé plánování. [3]

Tepelné elektrárny spalující uhlí v ČR zpravidla využívají energetické bloky s výkonem kolem 200–250 MW, přičemž rozdíly mezi jednotlivými elektrárnami spočívají zejména v počtu těchto bloků. Nejvíce tepelných elektráren je situováno v oblasti Krušných hor díky blízké těžbě hnědého uhlí. Mezi tyto elektrárny patří Tušimice, Počerady, Ledvice a Pruněřov. Další uhelné elektrárny se nacházejí například v Poříčí u Trutnova, v Dětmovicích na severní Moravě nebo v Hodoníně na jižní Moravě. [1]

Vedle uhelných elektráren jsou v České republice provozovány také paroplynové elektrárny, mezi které patří například Vřesová, Kladno a Počerady. Tyto elektrárny přispívají k diverzifikaci energetického mixu a mají nižší emisní zatížení ve srovnání s těmi uhelnými. [3]

1.2 Jaderná elektrárna

Stejně jako u tepelných elektráren je v jaderných elektrárnách využívána energie, teplo, které ohřívá vodu a mění ji na páru. Teplo je namísto spalováním, získáváno ze štěpné jaderné reakce. Při této reakci na radioaktivním prvku, nejčastěji uranu, dochází k tomu, že volný neutron pronikne do jádra atomu uranu, které se následně rozdělí na dvě přibližně stejně těžké části. Během tohoto procesu se uvolní 2 až 3 volné neutrony, které mohou dále štěpit další atomy uranu, čímž se spustí řetězová reakce. Jaderná energie se uvolňuje především ve formě tepla, které vzniká při zpomalování a zabrzdění částic vzniklých při štěpení. Toto teplo je využito podobně jako u tepelných elektráren, jak už bylo zmíněno. Část energie se však uvolňuje také ve formě nebezpečného jaderného záření, zejména neutronového toku. Toto záření je životu škodlivé, a proto je nutné zajistit účinnou ochranu okolí pomocí speciálních bezpečnostních opatření a bariér. [4]

Jaderné palivo, ve kterém probíhá štěpná reakce, se obvykle vyrábí ve formě palivových tyčí, kazet nebo koulí. Toto palivo obsahuje zpravidla přírodní uran, který je obohacen o vyšší podíl radioaktivního izotopu. Izotop představuje variantu chemického prvku s odlišným počtem neutronů v jádru atomu, což zvyšuje jeho štěpné vlastnosti. Palivo je umístěno v moderátoru, což je látka, která zpomaluje rychlost volných neutronů, aby mohly efektivněji způsobovat další štěpné reakce. Moderátorem může být voda, speciální chemické roztoky nebo pevné látky, jako je například grafit. Tento proces je klíčový pro udržení kontrolované řetězové reakce v jaderném reaktoru. [4]

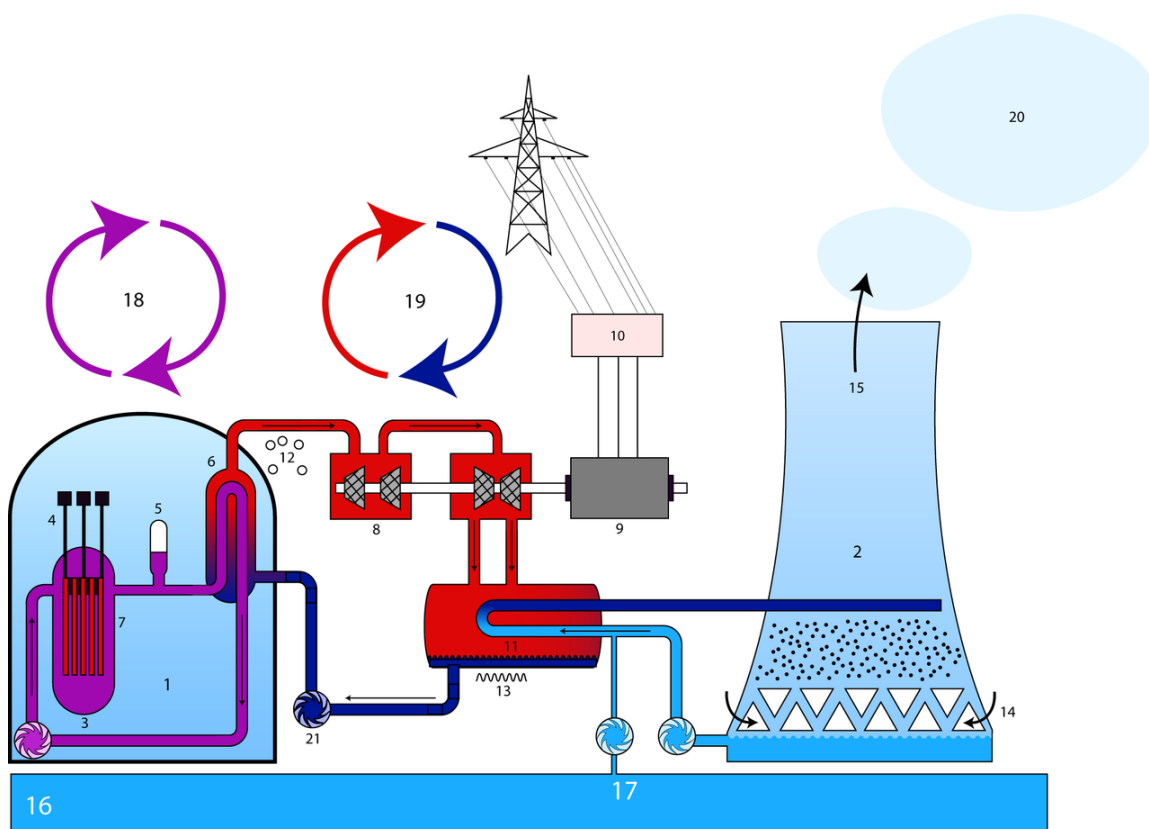
Řízení jaderné reakce je možné pomocí pohybu řídicích tyčí, které jsou vyrobeny z materiálů schopných pohlcovat volné neutrony, jako je kadmium nebo hafnium. Zasouváním těchto tyčí do reaktoru se počet volných neutronů snižuje, což reakci zpomaluje nebo zcela zastavuje. Naopak jejich vysouváním se umožňuje větší množství neutronů udržet řetězovou reakci, čímž se intenzita reakce zesiluje. Tento mechanismus je klíčový pro bezpečný provoz jaderného reaktoru a umožňuje přesné řízení jeho výkonu. [4]

Teplo, které vzniká při jaderné reakci, ohřívá kapalinu nebo plyn, nazývané chladivo. V jednookruhové elektrárně chladivo přímo pohání turbínu, zatímco v dvouokruhové elektrárně předává své teplo vodě ve výměníku, zvaném parogenerátor, což způsobí její přeměnu na páru, která následně pohání turbínu s generátorem. Dvouokruhová elektrárna tedy obsahuje primární okruh (radioaktivní), kde probíhá štěpná reakce, a sekundární okruh (neradioaktivní), kde je teplo přenášeno a využíváno pro výrobu elektřiny. [4]

Jaderné elektrárny se rozlišují podle typu chladiva a moderátoru, které jsou použity v reaktoru, a také podle toho, zda jsou jednookruhové, nebo dvouokruhové. Dvouokruhové elektrárny mají oproti jednookruhovým nižší účinnost, ale díky odděleným okruhům jsou bezpečnější. [4]

Jaderné reaktory se kromě elektráren využívají také v jiných aplikacích, například v jaderných ponorkách nebo lodích, kde parní turbína pohání nejen generátor, ale i lodní šroub.

Nejběžnějším typem jaderné elektrárny je elektrárna s tlakovodním reaktorem, která je dvouokruhová. Tento typ se používá také v českých jaderných elektrárnách Dukovany a Temelín. V primárním okruhu tlakové vody, zahřáté na přibližně 300 °C, je chladivo stlačeno tak, že při této teplotě nepřechází v páru. V parogenerátoru předává toto teplo vodě v sekundárním okruhu, která se mění na páru pohánějící turbínu. Voda v tomto typu reaktoru slouží také jako moderátor, který zpomaluje neutrony a podporuje štěpný proces. [4]



Obrázek 1 Schéma nejběžnějšího typu jaderné elektrárny s tlakovodním reaktorem [5]

Popis obrázku 1: 1) Reaktorová hala, uzavřená v nepropustném kontejnmentu; 2) Chladicí věž; 3) Tlakovodní reaktor; 4) Řídící tyče; 5) Kompenzátor objemu; 6) Parogenerátor – v něm horká voda pod vysokým tlakem vyrábí páru v sekundárním okruhu; 7) Aktivní zóna;

8) Turbína – vysokotlaký a nízkotlaký stupeň; 9) Elektrický generátor; 10) Transformační stanice; 11) Kondenzátor sekundárního okruhu; 12) Pára; 13) Kondenzát; 14) Přívod vzduchu do chladicí věže; 15) Odvod teplého vzduchu a páry komínovým efektem; 16) Řeka; 17) Chladicí okruh; 18) Primární okruh (voda pouze kapalná pod vysokým tlakem); 19) Sekundární okruh (červeně značena pára, modře voda); 20) Oblaka vzniklá kondenzací vypařené chladicí vody; 21) Čerpadlo. [5]

Po určité době, obvykle několika letech, je jaderné palivo „otráveno“ produkty štěpení, což znamená, že se stává vyhořelým. I když již nemůže být dále využíváno pro štěpnou reakci, zůstává stále vysoce radioaktivní. Vyhořelé palivo nemůže zůstat v této podobě, ale může být chemicky přepracováno na nové palivo nebo trvale uloženo na bezpečném místě, kde jeho záření nebude ohrožovat okolí. [4]

Dokud vyhořelé palivo stále vyzařuje nebezpečnou radiaci, což může trvat měsíce až roky, je uloženo v bezpečné oblasti jaderné elektrárny pod ochrannou vrstvou vody, která slouží k ochraně proti nebezpečnému záření. Po této době je palivo převezeno do meziskladu, kde může být uskladněno do doby, než bude přepracováno, nebo trvale uloženo na místě, které je navrženo k bezpečnému zajištění jeho dlouhodobé izolace od okolí. [4]

1.3 Elektrárny na biomasu

Dřevo je nejstarším využívaným zdrojem energie, který byl až do poloviny 18. století prakticky jediným palivem pro získávání tepelné energie. V 19. století však začala jeho spotřeba klesat, když fosilní paliva, jako je ropa, zemní plyn a uhlí, převzala dominantní roli. Ke konci 20. století, v rámci snahy o snížení vlivu lidské činnosti na změnu klimatu, získává dřevo a jiné formy biomasy opět velký význam, a to i v technicky vyspělých zemích, kde se stává součástí obnovitelných zdrojů energie. [6]

Biomasa je pojem, který zahrnuje veškerou organickou hmotu v biosféře, která je součástí koloběhu živin na planetě. To zahrnuje těla všech organismů, jako jsou živočichové, rostliny, bakterie, houby a sinice. [6]

Za energetickou biomasu jsou nejčastěji považovány rostliny. Spalování biomasy má oproti fosilním palivům téměř nulovou bilanci oxidu uhličitého, což znamená, že množství oxidu uhličitého uvolněného do ovzduší při spalování je přibližně stejné jako množství, které je zpětně absorbováno rostlinami během jejich růstu v zemědělských a lesních porostech

nebo na tzv. energetických plantážích. Kromě toho je u biomasy nižší obsah uvolňovaných oxidů síry. [6]

Biomasy se dělí podle vlastností a podle původu. Pokud lze biomasa spalovat přímo jedná se o suchá biomasa. Vlhké biomasy jsou tekuté odpady, které nelze spalovat přímo, ale musí se z nich vyrobit bioplyn v tzv. bioplynových stanicích. V těchto stanicích probíhá řízená fermentace neboli kvašení. Bioplynovou stanicí lze zásobovat spousty organickými materiály jako je hnůj, siláž, seno, sláma, zbytky jídla, nebo také mláto či kvasnice dovážené z pivovarů. Poslední rozdělení biomasy podle jejich vlastností jsou speciální, mezi které patří olejniny, škrobové a cukernaté plodiny. Z těchto biomas se získávají energetické látky jako je bionafta či líh. U rozdělení podle původu se dělí na biomasa pěstované pro energetické účely, kde se jedná hlavně o rychle rostoucí dřeviny nebo rostliny bylinného charakteru. Mezi takové rostliny patří topol, vrba, olše, konopí, šťovík, obiloviny, olejnaté rostliny či škrobové a cukernaté rostliny. Suroviny se před použitím musí zpracovat, tedy ze dřeva vyrobit štěpku a slámu či byliny nalisovat nebo vyrobit z nich brikety či granule. Původ biomas může být také odpadní, mezi které patří zbytky z rostlinné a živočišné výroby, zbytky potravin, nebo také zbytky z těžby a zpracování dřeva jako jsou větve, kůra nebo pařezy. [6]

Zpracování biomasy pro výrobu elektřiny zahrnuje spalování biomasy v kotlích. Spalováním biomasy nezískáváme pouze elektřinu, ale často se využívá i pro kogenerační výrobu, tedy kombinovanou výrobu elektřiny a tepla. Tento proces umožňuje efektivnější využití energie obsažené v palivu. Nejvyšší účinnosti biomasa dosahuje při výhradní produkci tepla, kde účinnost přesahuje 90 %. Naopak při čisté výrobě elektřiny je účinnost nižší, obvykle pod 50 %, protože velká část energie se ztrácí ve formě nevyužitého tepla. [6]

1.4 Vodní elektrárny

Vodní elektrárny využívají hydrologického cyklu, tedy stálého koloběhu vody na Zemi. Elektrárny mohou být konstruovány od nejmenších průtočných, o výkonech v řádu desítek kW, až po velkolepé přehradní elektrárny, o výkonech v řádu tisíců MW. Vodní elektrárny využívají energii proudící vody, která se skládá z kinetické a potenciální energie. Potenciální energie, někdy označovaná jako tlaková nebo polohová, je dána výškovým rozdílem hladin vody, tzv. spádem, a vzniká díky působení gravitace. Kinetická energie je naopak výsledkem pohybu vody a její velikost závisí na rychlosti toku. [7]

Princip vodní elektrárny spočívá ve vodě protékajícím přívodním kanálem, která předává svou kinetickou, respektive potenciální energii turbíně, která prostřednictvím společné hřídele pohání generátor. Ten díky principu elektromagnetické indukce přeměňuje rotační mechanickou energii na elektrickou energii. Celek složený z turbíny a generátoru se označuje jako turbogenerátor. [7]

Ve vodních elektrárnách se využívají různé typy turbín podle specifických podmínek a účelu dané vodní elektrárny. Kaplanovy turbíny s nastavitelnými lopatkami jsou nejčastější v podmínkách českých řek, protože jsou vhodné pro nízké spády a velké množství vody. Pro vysoké spády, dosahující až 500 metrů, se používají Peltonovy turbíny, které využívají trysky k přivádění vody na lopatky. V přečerpávacích elektrárnách je nasazována reverzní Francisova turbína, která může fungovat i jako čerpadlo. Malé vodní elektrárny často spoléhají na jednoduché horizontální turbíny, jako je Bánkiho turbína nebo modifikovaná Francisova turbína. [8]

Vodní elektrárny můžeme dělit jednak podle instalovaného výkonu, tak podle využívaného spádu, na kterém závisí například výběr turbíny. Nejznámější rozdělení je však podle využití vodního toku. [7]

1.4.1 Průtočné vodní elektrárny

Průtočné vodní elektrárny využívají přirozený průtok řeky, který není možné regulovat. Pokud objem vody přesáhne kapacitu, na kterou je elektrárna navržena, přebytečná voda je odvedena mimo turbíny bez využití. Kapacita elektrárny je určována takzvanou hltností turbíny. [7]

Tento typ elektráren se dále dělí na jezové a derivační. Jezové elektrárny využívají jez k navýšení hladiny vody a soustředění spádu. Tyto nízkotlaké elektrárny obvykle pracují se spádem mezi 10 a 20 metry. Naopak derivační elektrárny odvádějí část vody z hlavního toku pomocí přivaděče, což může být potrubí, kanál nebo štola. Voda je následně přivedena k turbíně a po průchodu elektrárnou vrácena zpět do řeky odpadním kanálem. Použití derivačního přivaděče umožňuje zkrátit část řeky, což zvyšuje spád a tím i efektivitu výroby energie. [7]

Průtočné elektrárny, které neumožňují regulaci průtoku vody, se využívají především k pokrytí základní spotřeby elektrické energie. Díky jejich provozu je zajištěna stálá výroba, i když není možné efektivně reagovat na změny v poptávce. [7]

1.4.2 Akumulační vodní elektrárny

Akumulační vodní elektrárny využívají přehradní hráze k zadržení vody a vytvoření potřebného spádu. Elektrárny se nacházejí buď přímo pod přehradou, nebo jsou s ní propojeny tlakovým přivaděčem. Tento typ vodní elektrárny využívá řízeného odběru z akumulací nádrže, a to podle potřeby elektrizační soustavy. [7]

Kromě výroby elektrické energie plní akumulací vodní elektrárny i další významné funkce. Stabilizují vodní toky, čímž pomáhají chránit okolí před povodněmi. Akumulační nádrže navíc často slouží jako zdroj pitné vody pro vodárny a poskytují technologickou vodu pro průmyslové a zemědělské účely. [7]

1.4.3 Přecherpující vodní elektrárny

Přecherpávací vodní elektrárny fungují jako akumulátory elektrické energie, které pomáhají pokrýt špičkové zatížení v elektrizační soustavě. Využívají dvě vodní nádrže umístěné v různých výškách. Do výše položené nádrže je voda čerpána, přičemž čerpání probíhá za využití přebytečné elektrické energie, obvykle během období vysoké výroby z obnovitelných zdrojů. Když je potřeba dodávat elektrickou energii, voda z horní nádrže proudí skrz turbínu do dolní nádrže, čímž generátor dodává energii do elektrizační soustavy. Akumulace vody může být zajištěna buď uměle, kdy veškerá voda je čerpána z dolní nádrže, nebo smíšeně, kdy část vody do horní nádrže přirozeně proudí z říčních toků. [7]

Za zmínku určitě stojí přecherpující vodní elektrárna Dlouhé stráně, která je lokalizovaná v Hrubém Jeseníku. Jedná se o největší českou vodní elektrárnu podle instalovaného výkonu (650 MW, o 170 MW více než elektrárna Dalešice), s největší reverzní vodní turbínou v Evropě a je také elektrárnou s nejvyšším spádem v České republice. [9]

1.5 Větrné elektrárny

Vítr se tvoří v atmosféře díky rozdílům v atmosférickém tlaku, které vznikají nerovnoměrným ohříváním zemského povrchu. Teplý vzduch stoupá vzhůru a jeho místo nahrazuje studený vzduch proudící z okolí. Rotace Země ovlivňuje směr větrných proudů, které dále mění krajinné prvky, jako je tvar terénu, přítomnost vegetace nebo vodních ploch. [10]

Větrná turbína přeměňuje energii proudícího vzduchu na mechanickou rotační energii, kterou generátor dále převádí na elektřinu. Listy rotoru jsou speciálně tvarované a využívají buď vztakovou sílu, nebo odporovou sílu, aby co nejefektivněji zachytily energii větru. [11]

Odporové turbíny, historicky starší typ větrných turbín, mají jednoduchý princip, ale nižší účinnost než modernější turbíny vztlakové. Fungují na základě rozdílu aerodynamického odporu lopatek vůči proudícímu větru. Tento rozdíl lze zajistit buď odlišným tvarem lopatek, které mají různý odpor v závislosti na směru větru (například miskovité lopatky), nebo řízeným natáčením lopatek podle polohy rotoru a směru větru. Druhý způsob je složitější, ale zajišťuje vyšší účinnost. [11]

Vztlakové turbíny jsou v současnosti nejrozšířenějším typem větrných turbín. Jejich princip spočívá ve využití aerodynamické vztlakové síly, která vzniká při obtékání lopatek vzduchem. Tato síla je generována díky specifickému tvaru profilu lopatek, podobně jako u křídel letadel, což umožňuje efektivní převod energie proudícího vzduchu na rotační pohyb. [11]

Větrné turbíny se podle osy otáčení dělí na horizontální (HAWT – horizontal axis wind turbine) a vertikální (VAWT – vertical axis wind turbine). [11]

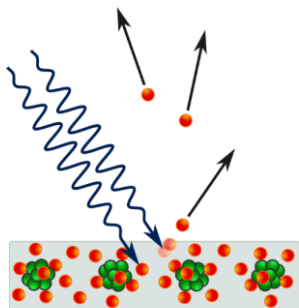
Horizontální turbíny vyžadují orientaci proti směru větru, což zajišťují buď směrové lopatky u menších zařízení, nebo větrné senzory a servo motory u větších elektráren. Často jsou vybaveny převodovým mechanismem, který zvyšuje rychlost pomaloběžného rotoru na úroveň vhodnou pro generátor. Tyto turbíny jsou nejčastěji využívány díky své vysoké účinnosti, která dosahuje přibližně 48 %. [11]

Vertikální turbíny mají výhodu v tom, že nepotřebují měnit směr vzhledem k větru, což je ideální v oblastech s proměnlivým větrem. Generátor a převodový mechanismus mohou být umístěny na zemi, což usnadňuje údržbu. Zároveň zabírají méně prostoru a lze je instalovat blíže k sobě bez výrazného aerodynamického ovlivnění, přičemž jsou méně hlučné. Jejich nevýhodou je vyšší pořizovací cena a nižší účinnost, která dosahuje přibližně 38 %. [11]

1.6 Fotovoltaické elektrárny

Fotovoltaická elektrárna (zkratkou FVE, nebo také fotovoltaika, solární či sluneční elektrárna) je energetické zařízení, které přeměňuje energii ze slunečního záření na energii elektrickou. Toho je docíleno díky fotoelektrickému jevu, který spočívá v uvolnění elektronů z obalu atomu křemíku, při ozařování P-N přechodů několika fotodiod tvořící fotovoltaický článek. Čím je intenzita slunečního záření větší, tím větší je rychlost vyraženého elektronu

a tím více se vyrábí elektřiny. Výsledný výkon je pak dán rozdílem mezi elektrony v pohybu a elektrony v klidovém stavu. [12]



Obrázek 2 Fotoelektrický jev [13]

Sdružením několika fotovoltaických článků vznikají fotovoltaické panely několika velikostí. Podle typu fotovoltaických článků mohou být panely monokrystalické, polykrystalické nebo amorfní. Liší se svými vlastnostmi a účinnostmi za různých světelných podmínek. [12]

Fotovoltaická elektrárna se dále skládá z elektrického střídače, který mění stejnosměrné napětí vyrobené fotovoltaickými panely na napětí střídavé, které napájí běžné spotřebiče a nachází se i v elektrické síti. Výběr elektrického střídače musí být přizpůsoben výstupnímu napětí, proudu a výkonu fotovoltaických panelů. Další součástí fotovoltaické elektrárny mohou být také akumulátory, které umožňují ukládání přebytečné energie pro pozdější využití. Mezi nejznámější akumulátory patří olovněné, nikl-kadmiové, nikl-metalhydridové nebo lithium-iontové, ale z ekonomických důvodů se v izolovaných zařízeních používají téměř vždy olovněné akumulátory. U domácích systémů pomáhají akumulátory pokrýt spotřebu večer nebo při výpadcích, zatímco u větších dodavatelů mohou sloužit ke stabilizaci dodávek energie do sítě. [12][14]

1.6.1 Domácí FVE

Jedna z nejdůležitějších rolí u investice do domácí fotovoltaické elektrárny hraje návratnost takové investice. V praxi to znamená, kolik dokáže domácnost ušetřit při nákupu elektrické energie. Hlavním problémem je, že fotovoltaická elektrárna vyrábí v časech, kdy je nejmenší spotřeba. Typický příklad je, že obyvatelé domácnosti jsou přes den v práci, či ve škole, kdy elektrárna nejvíce vyrábí, tudíž nepokrývá skoro žádnou spotřebu. V takovém případě je téměř veškerá energie prodávána do sítě za velmi nevýhodnou cenu. Návratnost bude tedy velmi nízká a zdoluhavá. Proto je potřeba k fotovoltaické elektrárně zavést také řízení, které bude určovat kdy a jak se bude v domácnosti spotřebovávat energie u spotřebičů, které lze

ovlivnit. Mezi ovlivnitelný spotřebič s velkou spotřebou energie patří ohřev teplé užitkové vody. Ten se většinou zapíná brzo ráno a také večer, kdy už, s výjimkou dlouhých letních dnů, elektrárna nevyrábí. Pokud však bude umožněn ohřev v časech, kdy elektrárna vyrábí a později večer bude jen pár stupňů přitopeno, ušetří to desítky kWh denně. Taková jednoduchá optimalizace dobře funguje za slunečného počasí, převážně v létě, kdy elektrárna vyrábí násobky celé spotřeby domácnosti. V zimě se ale poměr vyrobené a spotřebované elektřiny obrátí, a proto je potřeba další optimalizace, která využívá spotových cen energií. [15]

Spotová cena elektřiny je cena elektřiny, která platí v jednu konkrétní hodinu dne. To znamená, že každému, kdo nakupuje nebo prodává elektřinu na spotovém denním trhu, se cena změní dvacet čtyřikrát za den. Tyto ceny vypisuje regulační úřad a to vždy den dopředu. Výše spotové ceny v danou hodinu dne závisí na poptávce po elektřině a na tom, jaké zdroje elektřinu vyrábí, a jaká část z nich jsou obnovitelné zdroje, které spotovou cenu snižují. Poptávka bývá zpravidla nejvyšší ráno a večer, kdy fotovoltaické elektrárny nedodávají, nebo dodávají velmi málo, a proto bývá cena v těchto hodinách nejvyšší. Pokud to spotřebiče umožňují, a je možnost je spouštět výhradně v době, kdy je elektřina levnější, tedy přes den, je možné snížit až 30 % nákladů za elektřinu. S fotovoltaickou elektrárnou a akumulátory je možné dostat se až na 50 % úspor. V dnešní době už nejsou výjimkou ani záporné ceny elektrické energie. V praxi to znamená, že za elektřinu domácnost nezaplatí, ale naopak, dostane zapláceno. Takové ceny nastávají ve dnech, kdy je přebytek elektřiny v elektrické síti. Příkladem takového dne se zápornými cenami může být letní slunečný den, kdy fotovoltaické elektrárny vyrábí hodně elektřiny a lidé takový den tráví venku, což způsobí nízkou spotřebu. [16]

S využitím spotových cen při optimalizaci toku energie můžeme i v zimě velmi ušetřit, i když elektrárna vyrábí velmi málo. Za předpokladu, že ceny elektrické energie vypsané regulačním úřadem na další den budou v noci nízké a přes den velmi vysoké, může být příkladem optimalizace nabití akumulátoru a ohřev teplé užitkové vody v noci, za velmi malé částky, a přes den, kdy jsou ceny za elektrickou energii vysoké, bude spotřeba z části pokryta energií akumulovanou v akumulátoru. Přes léto mohou být naopak vybírány hodiny, kdy přebytečnou energii prodávat za co nejlepší cenu. [15]

Návratnost samotné fotovoltaické elektrárny by mohla trvat desítky let. S využitím zmíněných optimalizací toku energie v domácnosti může návratnost investice být i do pěti

let. Proto stojí určitě za zvážení, zda neinvestovat do řízení toku energie v domácnosti s fotovoltaickou elektrárnou.[15]

1.6.2 Dostupná řízení fotovoltaické elektrárny

Na českém trhu dnes působí řada firem, které nabízejí inteligentní řídicí jednotky určené pro fotovoltaické systémy. K nejvýznamnějším patří například Loxone, Cubee, SmartSpot či Infigy. Jednotlivá řešení se liší zejména poskytovanými funkcemi a úrovní kompatibility s ostatními komponentami v systému s fotovoltaickou elektrárnou, což je nutné zohlednit při výběru zařízení. V zahraničí navíc vznikají systémy, které do energetického managementu integrují prvky umělé inteligence, příkladem je platforma společnosti Huawei. Tato řešení cílí především na uživatele preferující plně automatizovaný provoz s minimální potřebou manuálního zásahu. [17][18]



Obrázek 3 Ukázka sledování toku energie od společnosti Loxone [19]

2 HARDWARE

Hardware představuje fyzickou část systému, která zahrnuje veškeré elektronické a mechanické komponenty nezbytné pro fungování zařízení. Tyto komponenty zajišťují interakci s okolním prostředím, zpracování vstupních a výstupních signálů a podporu softwarových funkcí. Správný výběr a konfigurace hardwaru jsou klíčové pro efektivní a spolehlivý provoz celého systému.

V této kapitole budou představeny jednotlivé hardwarové prvky, které tvoří součást bakalářské práce.

2.1 Mikropočítač

Mikropočítač, nebo také mikrokontroler, je velmi malý a levný počítač integrovaný do jednoho čipu, a ten následně integrován na vývojovou desku, většinou o velikosti dlaně. Je tvořen procesorem, pamětí, časovači, analogově-digitálním převodníkem, vstupní a výstupní piny a dalšími komunikačními rozhraními. Vše je uchyceno a propojeno na desce plošných spojů. U mikropočítače je nutno neplést si ho s mikroprocesorem. Oba pojmy, tedy předpona mikro, se vztahuje k fyzické velikosti zařízení. Zatímco mikropočítač znamená koncentrace mnoha funkcí na jeden čip o malé velikosti, s porovnáním například se stolním počítačem, a obsahuje mimo procesor také paměti a periferie, jak už bylo zmíněno. U mikroprocesoru jde pouze o samotný procesor, který je implementovaný v jednom integrovaném obvodu. Dnes už se dá považovat skoro každý moderní procesor za mikroprocesor, protože právě splňuje tuto definici, a používají se ve stejném významu. [20]

Mikropočítače dnes nacházejí široké uplatnění téměř ve všech typech elektronických zařízení. Příkladem jednoduchého použití je výstražné světlo na kolo, které je ovládáno jediným tlačítkem a nabízí několik režimů blikání či svícení. Dalšími běžnými aplikacemi jsou systémy v automobilech, jako je ABS, vstřikování paliva, ovládání oken, centrální zamykání nebo alarmy. Mikropočítače jsou také nedílnou součástí zařízení, jako jsou radiopřijímače, přehrávače, digitální fotoaparáty, tiskárny, mobilní telefony a mnoho dalších. V uvedených aplikacích jsou mikropočítače označovány anglickým termínem *embedded computers*, což v českém překladu znamená zabudované nebo vložené počítače. [20]

Zatímco dříve se pro programování mikropočítačů používal převážně assembler, dnes už pouze výjimečně. U většiny mikropočítačů byl assembler nahrazen jazykem C, C++, protože se jedná o nízko úrovněvý jazyk. Nízko úrovněvý jazyk znamená, že umožňuje přímou manipulaci s pamětí či registry. [20]

2.1.1 Raspberry Pi

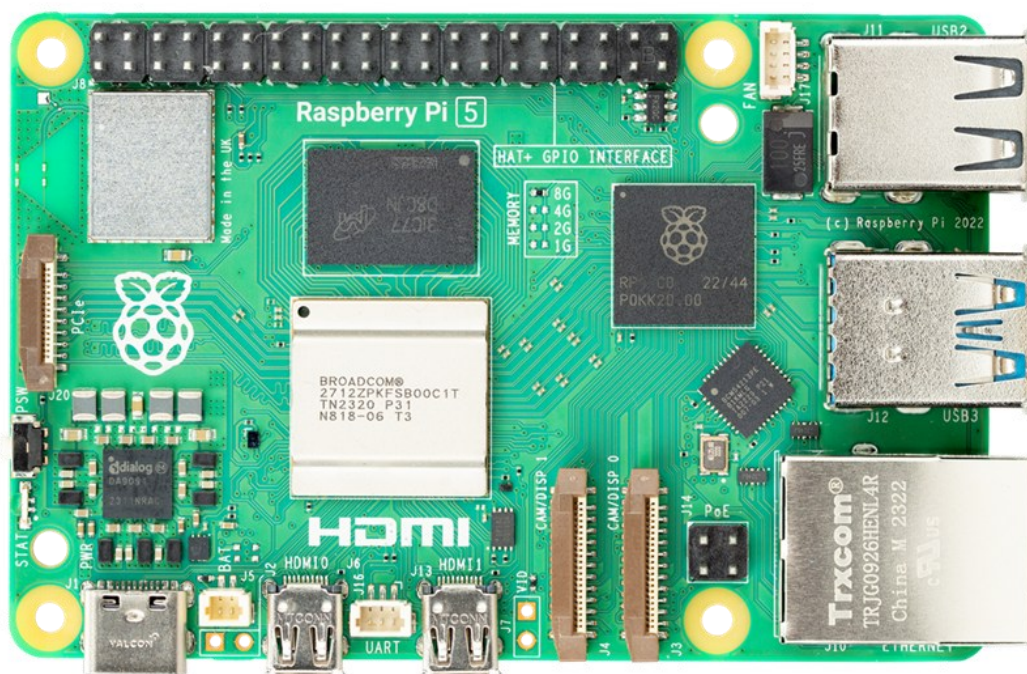
Raspberry Pi je jednodeskový počítač, o velikosti platební karty, který je srovnatelný se slabším stolním počítačem, hlavně díky HDMI konektorům pro připojení monitorů nebo USB konektorů pro připojení periferie jako klávesnice a myš. Běží na něm také vlastní operační systém od výrobce pojmenovaný Debian, který je variací Linuxu. Právě vlastním operačním systémem se nejvíce odlišuje od zmiňovaných mikrokontrolerů, ale také vyšším počtem periférií, externím uložištěm, grafickým či wifi čipem, ale i vyšší pořizovací cenou. Díky těmto rozdílům se Raspberry Pi řadí spíše do technologie system on chip (SoC), tedy česky systém na jednom čipu. Mikropočítače jsou určeny pro jednodušší aplikace jako jsou zmíněné zabudované systémy, kdežto SoC jsou spíše pro aplikace s vyššími požadavky na výkon i aplikace s vyšší složitostí. SoC může obsahovat i několik mikrokontrolerů na jednom čipu. [21][22]

Tabulka 1 Porovnání vybraných modelů Raspberry Pi [23][22][24]

Model	CPU	RAM	USB Ports	Ethernet	Wi-Fi	Bluetooth	HDMI	GPIO Pins
Raspberry Pi Model B	ARM1176JZF-S, 700 MHz	256 / 512 MB	2	10/100	N/A	N/A	1 x HDMI	26
Raspberry Pi 2	ARM Cortex-A7, 900 MHz	1 GB	4	10/100	N/A	N/A	1 x HDMI	40
Raspberry Pi 3	ARM Cortex-A53, 1.2 GHz	1 GB	4	10/100	802.11n	4.1	1 x HDMI	40
Raspberry Pi 3 B+	ARM Cortex-A53, 1.4 GHz	1 GB	4	10/100/300	802.11ac	4.2	1 x HDMI	40
Raspberry Pi 4	ARM Cortex-A72, 1.5 GHz	1 / 2 / 4 / 8 GB	4	10/100/1000	802.11ac	5.0	2 x micro-HDMI	40
Raspberry Pi 5	ARMv8.2-A76, 2,4 GHz	2 / 4 / 8 / 16 GB	4	10/100/1000	802.11ac	5.0	2 x micro-HDMI	40
Raspberry Pi Zero	ARM1176JZF-S, 1 GHz	512 MB	N/A	N/A	N/A	N/A	N/A	40
Raspberry Pi Pico	ARM Cortex-M0+, 133 MHz	264 KB	N/A	N/A	N/A	N/A	N/A	20

2.1.2 Raspberry Pi 5

Raspberry Pi prošlo několika vylepšeními. To bylo zapříčiněno nejen snahou zlepšování stávajících verzí, ale také obrovským posunem periferních i interních technologií. Původní Raspberry Pi byl vyvinut v roce 2012, což by s porovnáním tehdejších a dnešních technologií určitě nestačilo. Poslední, tedy 5. generace, vyšla na konci září roku 2023, která je oproti předchozí generaci (asi dva a půl roku staré) více než dvakrát výkonnější, a to díky vylepšení hardwaru i softwaru. Pro programování aplikací je nastaven výchozí jazyk Python. Programovací jazyk ale může být změněn, protože mikropočítač podporuje většinu nejznámějších programovacích jazyků, například C/C++ nebo Java. [25]



Obrázek 4 Raspberry Pi 5 [26]

Raspberry Pi je předurčený k vývoji aplikací v programovacím jazyce Python, ostatně jak už název mikropočítače napovídá. Python je vysokoúrovňový programovací jazyk, který je v dnešní době velmi populární pro svoji univerzálnost, ale také pro jednoduchost a přehlednost kódu, který se dobře čte i píše. Díky těmto vlastnostem se python čím dál tím více dostává do výuky na školách, nebo často bývá prvním jazykem samoučících se programátorů. Využití je od webových aplikací, až po vesmírné rakety. Python je vyvíjen

jako open source (software s otevřeným zdrojovým kódem) projekt, který je zdarma k instalaci na jakoukoli platformu, jako jsou Unix, MS Windows, macOS, android, či Linux. [27]

Tabulka 2 Parametry použitého Raspberry Pi 5 [23]

CPU	čtyřjádrový ARMv8.2 - A76, 2.4 GHz, 64 b
RAM	4 GB LPDDR4X SDRAM
HDMI	2x mikro HDMI
USB	2x 3.0 USB + 2x 2.0 USB
Ethernet	10/100/1000
Wi-Fi	802.11ac
Bluetooth	5.0
GPIO	40 pinů
Další konektory	2x 4 - line mini MIPI CSI/DSI konektory pro připojení kamery nebo obrazovky
Paměť	slot pro mikro SD karta
Napájení	USB – C (5 V / 5 A)

2.1.3 Vývojové prostředí Visual Studio Code

Visual Studio Code (VS Code) je bezplatný a výkonný editor zdrojového kódu, který je dostupný pro operační systémy Windows, macOS, Linux a Raspberry Pi OS. Nabízí rychlý start, nízké nároky na systém a rozsáhlou sadu funkcí vhodných pro moderní vývoj. Kromě klasické desktopové verze je možné VS Code využívat i ve webovém rozhraní. Editor je přednastaven pro práci s jazyky JavaScript, TypeScript a Node.js, přičemž další podporu lze snadno rozšířit prostřednictvím rozšíření. K dispozici jsou tisíce doplňků pro programovací jazyky jako C++, C#, Java, Python, PHP nebo Go, ale také pro prostředí jako Docker, Kubernetes či cloudové platformy Amazon Web Services, Microsoft Azure a Google Cloud Platform. [28]

VS Code nabízí funkce jako inteligentní doplňování kódu (IntelliSense), grafické ladění, kontrolu syntaxe (linting), práci s více kurzory, nápovědu parametrů funkcí, navigaci v kódu, refaktoring a integrovanou správu verzí pomocí Git. Mnoho těchto funkcí vychází z technologií používaných ve Visual Studiu. Vývojáři oceňují i podporu vzdáleného vývoje, což umožňuje pracovat například s kontejnery nebo servery bez nutnosti instalovat vše lokálně. [28]

Rozšíření Visual Studio Code Remote – SSH umožňuje připojit se k libovolnému vzdálenému počítači, virtuálnímu stroji nebo kontejneru, který má spuštěný SSH server, a pracovat se vzdálenými soubory tak, jako by byly lokální. Díky tomu lze plně využít všechny funkce editoru VS Code bez nutnosti mít zdrojové kódy uložené na lokálním počítači. Po navázání připojení se na vzdáleném systému automaticky nainstaluje služba VS Code Server, která běží nezávisle na případné instalaci VS Code na daném zařízení a umožňuje spouštění příkazů i rozšíření přímo na vzdáleném stroji. [29]

Rozšíření tvoří důležitou součást ekosystému VS Code. V oficiálním katalogu rozšíření (Marketplace) je dostupných více než 38 000 rozšíření, přičemž nejpopulárnější doplňky, jako například pro Python, mají desítky milionů instalací. Tato rozšíření mohou přidávat nejen podporu dalších jazyků, ale také například nástroje pro práci s Jupyter notebooky, kontejnery nebo cloudovými službami. [28]

Visual Studio Code používají miliony vývojářů po celém světě, včetně zaměstnanců firem jako Google, Facebook nebo Microsoft. Oblíbený je nejen mezi webovými vývojáři, ale také mezi datovými analytiky a vědci, kteří využívají možnosti rozšíření pro Python, R a Jupyter notebooky. Oblíbenost VS Code pramení z kombinace jednoduchého prostředí, rozsáhlých možností přizpůsobení a výkonných vývojových nástrojů. [28]

2.2 PLC

PLC neboli programovatelný logický automat je dnes relativně malý průmyslový počítač uzpůsobený pro potřeby a řešení automatizačních úloh v reálném čase, s co nejkratší odezvou. Úlohy jsou převážně logického typu, hlavně u starších nebo menších úloh, proto také název „logický“. Dnes už ale nejsou výjimkou také úlohy implementující složité diskrétní regulátory. Mezi nejčastější aplikace patří výrobní procesy, jako jsou montážní linky, stroje nebo také robotická zařízení. Hlavními cíli, které musí PLC splňovat jsou vysoká spolehlivost, vysoká opakovatelnost a jednoduchost při programování, rozšiřování či údržbě. Při většině aplikací je požadován nepřetržitý provoz, tzn. že PLC musí být k tomuto přizpůsobeno. Vzhledem k lokalitě použití (nejčastěji průmysl) musí být PLC odolné proti vibracím a nárazům, prachu a vlhkosti, teplotním extrémům, nebo také proti elektromagnetickému rušení. Také díky těmto vlastnostem je PLC schopno pracovat nepřetržitě, a to několik let až desítky let. [30]

2.2.1 Dělení PLC

Základní dělení PLC je podle jejich konstrukce:

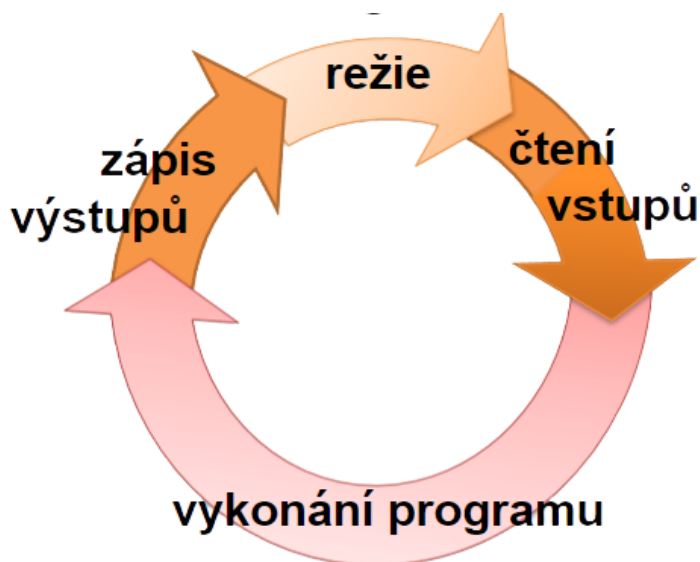
- **Kompaktní** – PLC tohoto typu jsou situována do jednoho modulu, který obsahuje CPU, disponuje pouze jednotkami vstupů a výstupů a základní podporu komunikace. Hlavní nevýhodou je právě limit počtu vstupů a výstupů, a je tedy vhodný pouze pro malé aplikace s nízkým počtem vstupů a výstupů. Výhodou tohoto typu PLC je nízká pořizovací cena. [31]
- **Modulární** – Hlavním rozdílem modulárního typu PLC oproti typu kompaktnímu, je možnost přidávání modulů pro různou aplikaci. Srdce takového PLC často bývá pouze procesor bez žádných vstupů a výstupů, či jiné komunikace, dokonce i bez napájení. Pro napájení celého PLC je potřeba dokoupit napájecí modul, a taktéž moduly pro komunikaci s jednotlivými periferiemi. Pro připojení modulů mezi sebou, mají vyvedenou interní sběrnici v rámci PLC. Výhodou takového typu PLC je velká škálovatelnost, možnost zapojení velkého počtu vstupů a výstupů, a je tedy vhodný pro aplikace větších rozměrů, jako jsou celé výrobní stroje. [31]

Kompaktních PLC již ale ubývá, protože výrobci i těmto typům PLC přidávají omezenou modulárnost. [32]

Z hardwaru musí každé PLC obsahovat CPU, paměť, vstupní a výstupní rozhraní, komunikační rozhraní s programátorem či obsluhou a také napěťový zdroj, který převede střídavé napětí na stejnosměrné a u většiny PLC také na pracovních 24 V. Takový zdroj může být externí či zabudovaný přímo v modulu PLC. Všechny hardwarové součásti jsou mezi sebou propojeny systémovou sběrnici, která zařizuje jejich komunikaci mezi sebou. [32]

Způsob zpracování programu je odlišný než, u klasických počítačů, či mikropočítačů. Základní vlastností je cyklické provádění programu. Jedna otočka cyklu trvá řádově milisekundy a závisí na obsaženém počtu instrukcí. Doba vykonávání jedné otočky cyklu bývá parametrem výkonu CPU, a tedy i parametr, podle kterého bude nakupující vybírat typ PLC. Ačkoli se často uvádí rozmezí nebo pevně stanovená doba trvání jednoho cyklu, složitější operace mohou tuto dobu výrazně prodloužit, případně vyžadovat více cyklů pro své dokončení. [32]

Každá otočka cyklu začíná inicializací systémových a časových proměnných, následně probíhá načtení vstupů PLC, poté je vykonáván program a výsledné hodnoty jsou zapsány na výstupy. Na závěr cyklu se provádějí režijní operace, například aktualizace systémových proměnných a příprava procesů na další otočku cyklu. [32]



Obrázek 5 Otočka cyklu PLC [33]

U PLC je běžné, že uživatelský program nepracuje přímo s hodnotami vstupů a výstupů. Místo toho se při čtení vytvoří tzv. „obrazy“ těchto hodnot v paměti, čímž je zajištěna synchronizace dat s během programu. Tento přístup má klíčový význam pro předcházení hazardním stavům, které mohou vzniknout, pokud by během provádění programu došlo ke změně některého vstupu. Tím by mohlo být ovlivněno správné vyhodnocení výpočtů, což by v automatizačních úlohách mohlo vést k nežádoucímu chování. [32]

2.2.2 Vývojové prostředí Multiprog

Programovatelné automaty se vyznačují různými vývojovými prostředími, která si často vytvářejí a spravují sami výrobci PLC. Pro sjednocení těchto prostředí byla vytvořena mezinárodní norma IEC 61131-3, která definuje pět programovacích jazyků: Structured Text (ST), Ladder Diagram (LD), Function Block Diagram (FBD), Instruction List (IL) a Sequential Function Chart (SFC). Vývojové prostředí Multiprog, vyvinuté společností KW-Software GmbH, tuto normu podporuje. Umožňuje kombinaci všech uvedených jazyků v rámci jednoho projektu, což usnadňuje vývoj složitějších aplikací a zvyšuje přehlednost kódu. Součástí prostředí jsou také nástroje pro konfiguraci hardwaru, editaci programového kódu, ladění a vizualizaci běhu programu. [36]

CRC apod.) jsou uspořádána v registrech, které mají svoji unikátní adresu, na kterou se při komunikaci odkazuje. Aby bylo zařízení snadno implementovatelné, dodává každý výrobce i tabulku registrů. V tabulce jsou zaznamenány adresy jednotlivých registrů, jejich velikost a data, která jsou v nich uložena. Tento protokol je určen pro komunikaci s nízkou rychlostí a krátkým dosahem. [34]

Další variantou Modbus komunikace je Modbus TCP/IP. Protokol funguje na stejném principu jako Modbus RTU, tedy „dotaz-odpověď“, s lehkou úpravou podoby příkazů. Zároveň komunikace neprobíhá na sériové lince ale v síti Ethernet a varianta je navržena pro vysokorychlostní komunikaci pro aplikace náročné na data, řízení v reálném čase a rozsáhlé průmyslové sítě. [35]

II PRAKTICKÁ ČÁST

3 ANALÝZA SOUČASNÉHO STAVU SYSTÉMU

Domácnost je vybavena fotovoltaickým systémem, který umožňuje efektivní využití sluneční energie pro generování energie elektrické. Systém zahrnuje fotovoltaické panely instalované na střeše domu a střídač s dvěma akumulátory situované v garáži. Dále je v domě instalovaný elektrický bojler jako přídavná zátěž pro využití přebytečné energie a PLC pro základní regulaci přebytečné energie.



Obrázek 7 Fotovoltaické panely na domě

Fotovoltaické panely jsou situovány na jižní stranu střechy, tak aby bylo nejvíce využito sluneční energie. Panelů je celkem 14 kusů a každý z nich má výkon 550 Wp. Celkový výkon fotovoltaické elektrárny je tedy 7,7 kWp. Jedná se o monokrystalické panely s účinností 21,2 %, které mají velmi příznivý poměr cena / výkon. [37]

Elektrický střídač řady X3-Hybrid čtvrté generace od výrobce Solax power, slouží pro přeměnu stejnosměrného napětí vyrobeného fotovoltaickými panely na napětí střídavé a ukládá jej do akumulátoru. Důvodem přeměny napětí ze stejnosměrného na střídavé je, že většina spotřebičů, ale také elektrická síť, funguje na střídavé napětí a také spínání stejnosměrného napětí je problematictější než spínání střídavého napětí. Střídač má 4 pracovní režimy: maximalizace vlastní spotřeby, priorita přetoku do sítě, režim zálohy (UPS) a nouzový zdroj energie (EPS). [39]

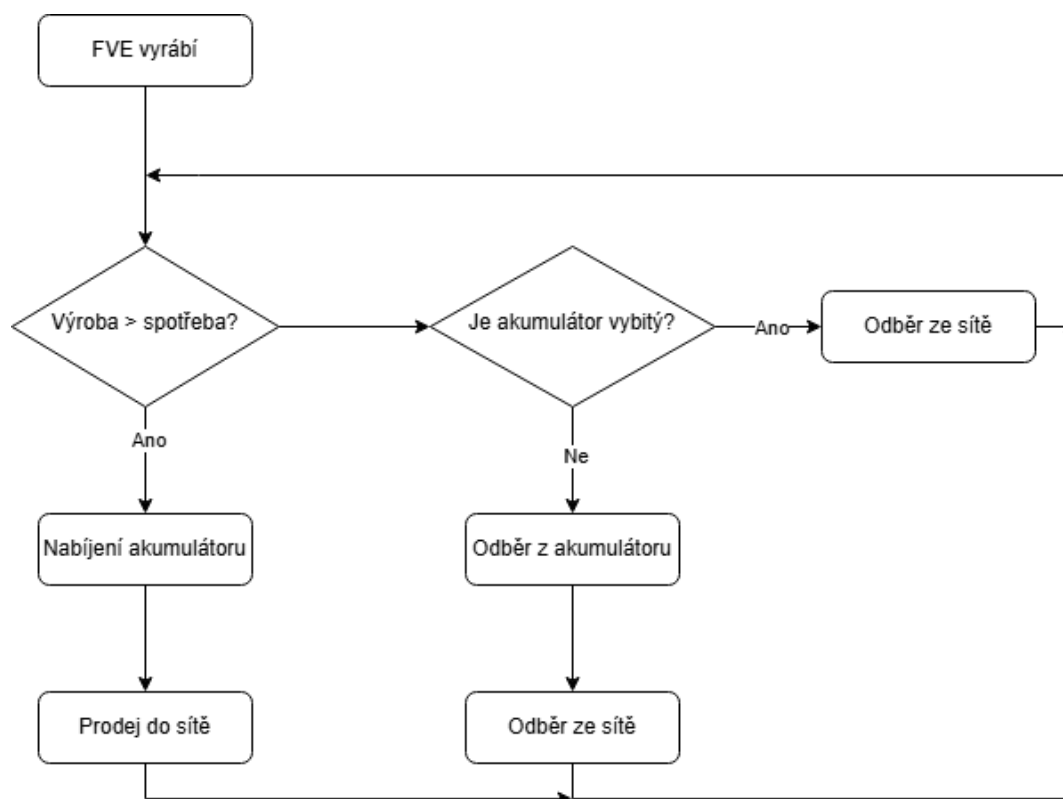
Použité akumulátory jsou Solax Triple HV battery s kapacitou 5,8 kWh, tedy celkovou kapacitou 11,6 kWh. Jedná se o technologii Li-ion (LFP) s minimální životností 6000 cyklů. [38]

PLC je v systému využito pro PWM řízení elektrického bojleru při přebytečné vyrobené energii. Jedná se o OMC8000 od českého výrobce Orbit Merret. Obsahuje 3 digitální vstupy a 6 univerzálních vstupů, které umožňují připojení signálů, jako jsou signály impulzní do 30 V, analogový (napěťový do 30 V, proudový do 20 mA, odporový do 3900 Ω) nebo také analogový signál z Pt100 či Pt1000. Dva z těchto univerzálních vstupů mohou být také použity pro komunikaci přes sběrnici RS485, a to pomocí protokolu ASCII nebo MODBUS – RTU. Dále PLC obsahuje 5 digitálních výstupů, displej a sběrnici pro rozšiřující moduly.

3.1 Současné řízení systému

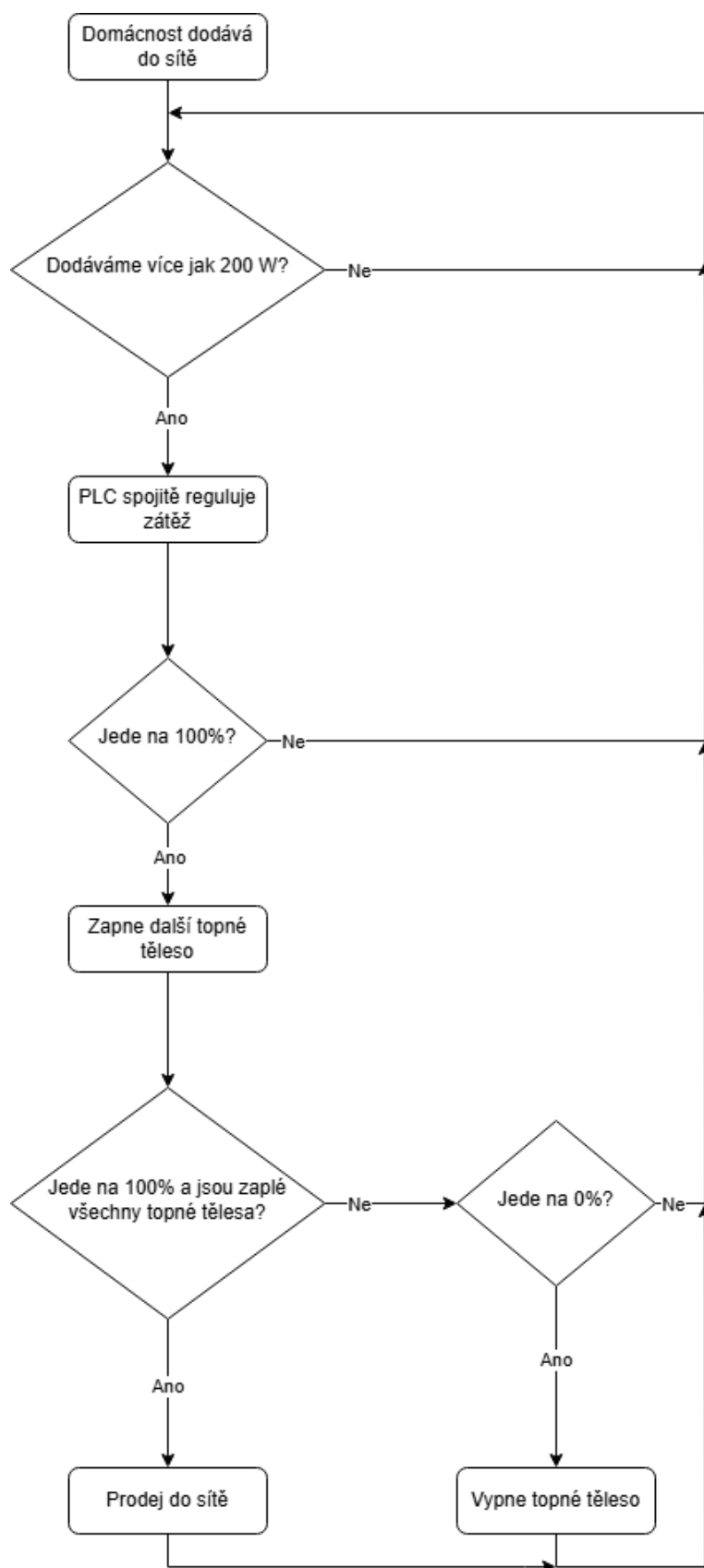
Systém obsahuje chytrý střídač, který je nastaven v pracovním režimu maximalizace vlastní spotřeby. To znamená, že vyrobená energie je přímo spotřebována domácností. Zde může nastat několik situací:

- 1) Vyrobená energie je větší než spotřeba domácnosti: v takovém případě se nespotebovanou energií nabíjí akumulátor, a pokud je i akumulátor nabitý, posílá se energie do sítě.
- 2) Vyrobená energie je menší než spotřeba domácnosti: energie, která schází k pokrytí spotřeby se musí odebírat ze sítě, ale pokud je akumulátor nabitý a je aktivní vybíjecí perioda, odebírá se potřebná energie napřed z akumulátoru a až následně ze sítě.
- 3) Solární energie není dostupná (typicky v noci): spotřeba je primárně kryta nabitým akumulátorem a následně ze sítě. Pokud je akumulátor vybitý je spotřeba pokrytá energií ze sítě a ze sítě se také může akumulátor dobíjet. [39]

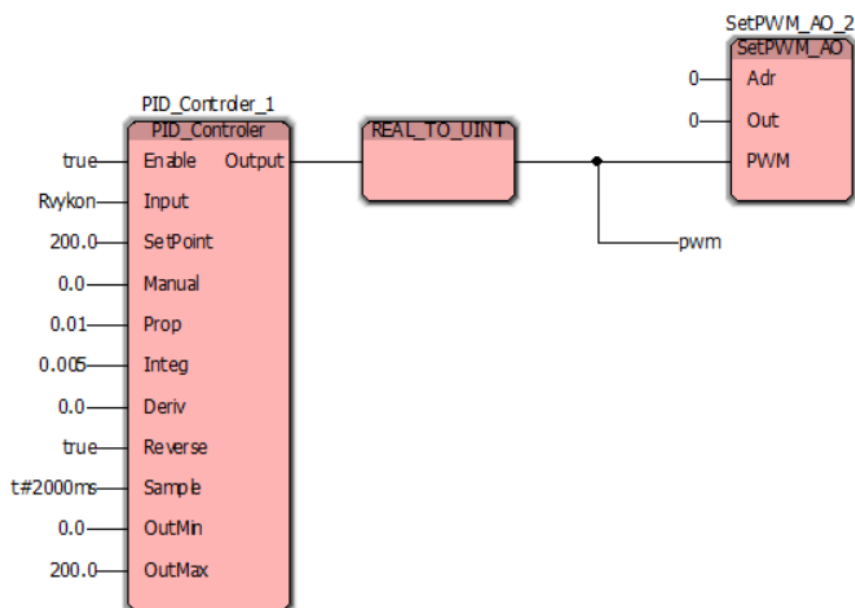


Obrázek 8 Vývojový diagram ukazující práci chytrého střídače s energií

Jako dodatečné řízení pro využití nespotřebované solární energie bylo přidáno PLC OMC8000, ke kterému je přes SSR relé připojen elektrický bojler o objemu 160 litrů se třemi topnými tělesy o výkonu 1 kW. PLC komunikuje s chytrým střídačem pomocí sběrnice RS485 a protokolu MODBUS – RTU, ze kterého získává hodnotu výkonu, který domácnost dodává či odebírá ze sítě, a také teplotu akumulátoru. Získaný výkon je pak předán PID regulátoru, který reguluje příkon boileru prostřednictvím PWM modulovaného řídicího signálu. PWM pak řídí SSR relé, které spíná topné těleso boileru a tím plynule reguluje jeho výkon v závislosti na přetocích do sítě. Aby se zamezilo překmitům, které by zapříčinily odběr ze sítě, byla zvolena rezervní hodnota přetokového výkonu na 200 W, tedy virtuální nula, od které začíná PID regulátor regulovat. Hodnoty PID regulátoru byly nastaveny experimentálně a to: proporcionální složka 0,01, integrační složka 0,005 a derivační složka na 0. Jedná se tedy pouze o PI regulátor.

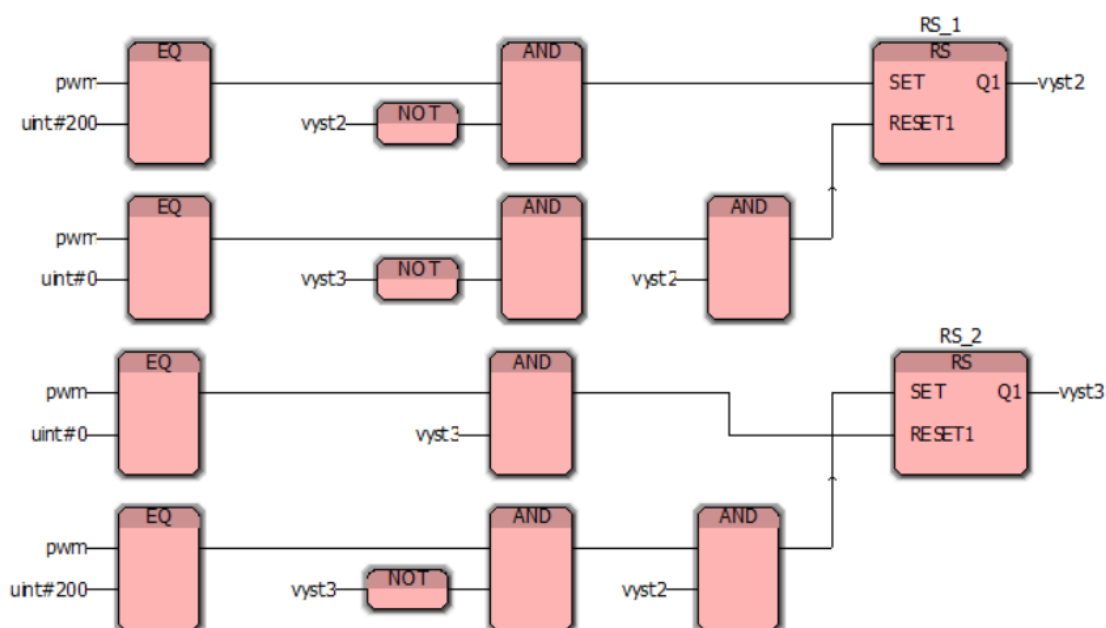


Obrázek 9 Vývojový diagram regulace výkonu bojleru



Obrázek 10 Program pro regulaci výkonu bojleru pomocí PID regulátoru

Elektrický bojler je vybaven třemi topnými tělesy. První z nich je řízené PWM modulovaným signálem pomocí SSR relé, umožňující plynulou regulaci výkonu v rozsahu 0–1 kW. Zbývající dvě topná tělesa mají pevně daný výkon 1 kW a jsou ovládána pouze binárním spínáním (zapnuto/vypnuto). Ty jsou ovládána jednoduchým programem, který sepne přídatné topné těleso, pokud to spojitě řízené pracuje na plný výkon.



Obrázek 11 Program pro spínání přídatných topných těles

Jak bylo zmíněno, z registru střídače je také získávána teplota akumulátoru. Aby nedocházelo k rychlé degradaci akumulátoru vlivem nízkých teplot, bylo instalováno externí topné těleso, které je připojeno na jeden výstup PLC. V PLC je pak naprogramován jednoduchý dvoupolohový regulátor s hysterezí 6 °C, který se spouští při poklesu teploty pod 12 °C.

```
Temperature:=BYTE_TO_INT(BTemperature);  
  
IF Temperature <= 12 THEN  
    BatteryHeating := true;  
  
ELSIF Temperature >=18 THEN  
    BatteryHeating := false;  
  
END_IF;
```

3.2 Optimalizace řízení systému

Cílem optimalizace je zefektivnit tok energie tak, aby se maximalizovalo využití vlastní vyrobené elektřiny a minimalizovaly náklady na odběr ze sítě, čímž se sníží celkové výdaje za elektřinu. Toho bude docíleno rozšířením současného řídicího systému mikropočítačovou jednotkou, která bude pracovat s vícero faktory, ovlivňující tok energie, a nejen pouze s přetoky výkonu do sítě. Mezi tyto faktory patří průměrná spotřeba energie domácnosti, spotové ceny, predikce slunečního osvětlení a volná kapacita akumulátoru. Díky těmto faktorům je možné určit, jaká bude přibližná celková spotřeba na další den, kolik energie elektrárna vyrobí a jaké budou ceny za energie.

Data, jako jsou denní výroba elektrické energie, energie odebrána ze sítě za den, energie dodána do sítě za den, množství energie nabitě či vybitě z akumulátoru za den, jsou získávána ze střídače přes Modbus TCP pro výpočet denní spotřeby. Tato data jsou získávána jednou za den, a to těsně před půlnocí dříve, než budou resetována pro další den. Dále jsou přes tuto komunikaci získávány přetoky do elektrické sítě nebo teplota akumulátoru, které slouží pro vizualizaci a ověření správné funkce systému. Přetoky do sítě jsou získávány jednou za 2 sekundy a teplota akumulátoru jednou za minutu. Pro výpočet spotřeby na další den je využita hodnota aktuální kapacity akumulátoru, která je také získávána přes tuto komunikaci a je čtena jednou denně.

Předpověď slunečního osvitu je získáván ze služby PV Forecast, která tuto předpověď poskytuje pro následující 24 až 48 hodin pro libovolný bod v České republice. Tato předpověď je poskytována ve W/m^2 pro každou hodinu a je vztažena k vodorovné rovině. Pro získání předpovědi výroby elektrárny je použit tento výpočet:

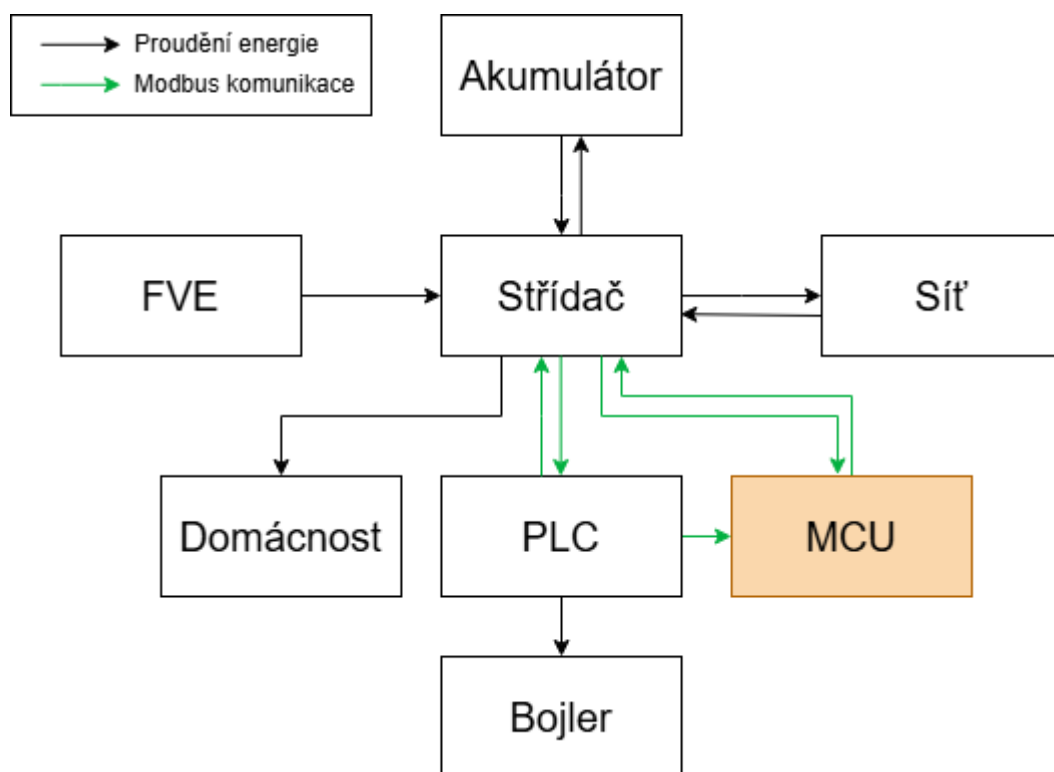
$$P = \text{osvit} * A * \eta * n \quad (1)$$

kde: P [W] ... Predikce výroby elektrárny
 osvit [W/m^2] ... Predikce světelného osvitu
 A [m^2] ... Plocha solárního panelu (2,584)
 η [-] ... Účinnost fotovoltaiického panelu (0,212)
 n [-] ... Počet panelů v elektrárně (14)

Tato data jsou získávána jednou denně, a to hned po půlnoci, kdy jsou zpřesněny predikce pro následující den.

Spotové ceny jsou získávány z webu ote-cr.cz, který kolem čtrnácté hodiny vypisuje po hodinách ceny elektrické energie pro další den. Tyto ceny jsou získávány jednou denně. Veškeré zmíněná data získávané z různých zdrojů jsou ukládána do databáze InfluxDB, pro vizualizaci, kontrolu a kontrolu správnosti fungování řízení.

Výpočet přibližné spotřeby pro další den program počítá z průměru několika posledních spotřeb pro daný den, zapsané v databázi, z chybějící energie v akumulátoru, a ještě je k ní připočítána bezpečnostní konstanta. Tato spotřeba je odečtena od celkové predikce výroby na daný den, a z toho nám vyjde přebytečná energie, která bude poslána do elektrické sítě, nebo naopak energie zbývající, která bude dodatečně z elektrické sítě odebrána. Cílem je tuto přebytečnou energii prodávat za co nejvyšší ceny. Proto budou stahovány zmíněné spotové ceny, které budou porovnávány s hodinovou predikcí osvitu, tak aby se přiřadili předpokládané energetické přetoky do hodin s nejvyšší cenou. Zároveň v těchto hodinách s prioritou exportu energie do sítě bude zakázáno nabíjení akumulátoru a spínání přídatné zátěže. Pro hodiny s nejvyšší spotovou cenou, a zároveň při výrobě elektřiny, bude zakázáno nabíjení akumulátoru, a také spínání přídatné zátěže. Další nedílnou součástí řízení je zákaz jakéhokoli exportu energie do sítě při záporných spotových cenách.



Obrázek 12 Blokový diagram systému, včetně implementovaného mikropočítače

4 POPIS REALIZACE

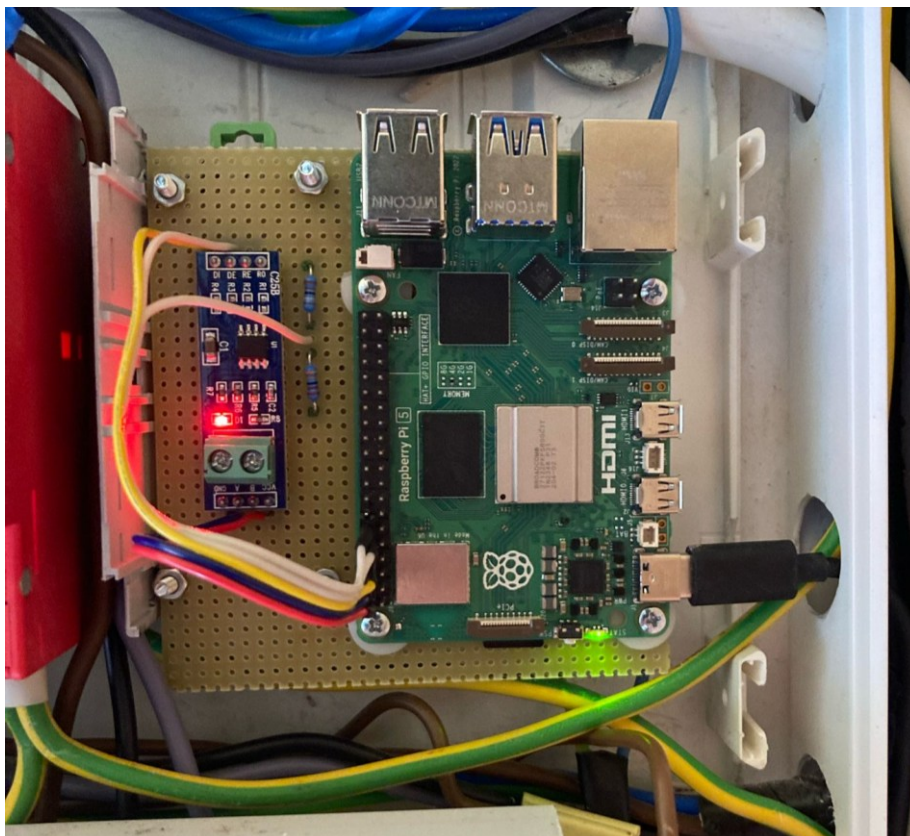
Tato kapitola popisuje kompletní průběh realizace bakalářské práce, zahrnující všechny klíčové fáze od počátečního výběru mikropočítače až po vývoj programu a vizualizaci.

4.1 Výběr mikropočítače

Hlavním řídicím prvkem stávajícího systému je hlavně střídač, ke kterému je přes Modbus RTU komunikaci připojené PLC pro spínání přídavné zátěže. Požadované rozšíření řízení systému v PLC není možné vzhledem k tomu, že neumí stahovat a zpracovávat data z internetu jako jsou spotové ceny a predikce osvitu. Proto bylo nutné navrhnout další řídicí jednotku, tedy mikropočítač. Raspberry Pi 5 je pro tuto aplikaci vhodný zejména díky svému poměru cena/výkon. Disponuje vysokou rychlostí procesoru s širokou nabídkou velikostí RAM pamětí, má zabudované WiFi komunikační rozhraní pro připojení k internetu a zároveň i několik GPIO pinů pro připojení periférií. Pro tuto práci bylo zvoleno Raspberry Pi 5 4 GB RAM, které pro tuto aplikaci stačí s dostatečnou rezervou.

4.2 Instalace mikropočítače

Pro umístění mikropočítače bylo vymezeno místo v rozvaděči, kde se nachází veškeré technické vybavení pro chod fotovoltaické elektrárny: řízení přídavné zátěže (PLC), jištění přídavné zátěže a také jištění elektrických zahradních zařízení, jako je například čerpadlo umístěné ve studni. Mikropočítač byl umístěn na univerzální desku tištěných spojů, kde byl přidán napěťový převodník pro komunikaci přes sériovou linku RS485. Jelikož výstupní napětí z převodníku, tedy převedené napětí ze sběrnice, dosahuje hodnoty 5 V, zatímco GPIO piny mikropočítače Raspberry Pi pracují s logickými úrovněmi 0 - 3,3 V, bylo nezbytné zařadit napěťový dělič, který zajistí přizpůsobení napěťové úrovně vstupním parametrům Raspberry Pi. Tento napěťový dělič je tvořen dvěma rezistory a to 1,8 k Ω a 3,3 k Ω . Hodnoty odporu rezistorů byly navrženy tak, aby příliš proudově nezatěžovaly výstup linkového budiče a zároveň aby bylo napětí stabilní. Při vysokých odporech by mohl GPIO vstup napětí ovlivňovat a může nastat pomalá odezva či zkreslení. Celá deska univerzálních plošných spojů byla pak umístěna na zbytek krytu jističe tak, aby bylo možné celou desku umístit do rozvaděče na DIN lištu. Výsledný modul s mikropočítačem lze vidět viz Obrázek 13.



Obrázek 13 Raspberry Pi a napěťový převodník pro připojení ke sběrnici umístěné v rozvaděči

4.3 Operační systém mikropočítače a vzdálený přístup

Pro nainstalování operačního systému bylo nutné dokoupit vysokorychlostní (class 10) SD kartu, která slouží jako hlavní uložisko mikropočítače. Poté je potřeba SD kartu naformátovat a nahrát na ni operační systém. K tomu je nejvhodnější použít oficiální software Raspberry Pi Imager, který je dostupný na oficiálních stránkách raspberrypi.com. Tento software umožňuje nastavit jméno i heslo, nastavení WiFi připojení a také nabízí nainstalování SSH souborů pro vzdálený přístup.

Po vložení SD karty do mikropočítače a připojení napájení se systém nabootoval a jeho funkčnost byla ověřena pomocí příkazu ping z jiného zařízení připojeného k síti. Následně po vzdáleném připojení k mikropočítači z VS Code byl zaktualizován systém a ověřena verze pythonu pomocí příkazů:

```
sudo apt update && sudo apt upgrade -y  
python3 -version
```

4.3.1 Připojení VS Code pro vzdálený přístup

Pro využívání vzdáleného přístupu ve vývojovém prostředí Visual Studio Code je nutné nainstalovat balíček *SSH – Remote*. Po instalaci je postup následující:

- Do vyhledávacího pole na horní liště programu zadat příkaz:
>Remote-SSH: Add New SSH Host...
- Zvolíme požadované zařízení podle jména, v tomto případě:
ssh admin@raspberrypi.local
- A v poslední řadě zvolíme ssh soubor zařízení, v tomto případě: *~/.ssh/config*
- Teď už máme zařízení přidané a pro připojení zadáme do vyhledávacího pole příkaz: *>Remote-SSH: Connect Current Window to Host...* a po zadání hesla, které jsme si zvolili při instalaci operačního softwaru, jsme vzdáleně připojeni k zařízení

4.4 Vývoj programu mikropočítače

V této kapitole jsou popsány veškeré vytvořené dílčí programy v mikropočítači, které jsou následně využity pro chod hlavního programu. Je zde vysvětleno k čemu slouží i detailní popis funkčnosti zdrojového kódu.

4.4.1 Stahování spotových cen

Funkce stahování spotových cen využívá knihovny *requests*, *datetime* a *BeautifulSoup* z knihovny *bs4*. Knihovna *request* slouží k připojení programu na webovou stránku pomocí http požadavku. Http je protokol, který umožňuje komunikaci mezi klientem (např. webovým prohlížečem) a webovým serverem. Knihovna *request* zde tedy zastupuje webový prohlížeč. Jelikož stahování spotových cen musí probíhat automaticky každý den, musí se měnit i datum v url adrese, ze kterých jsou data stahována. Zde je využito knihovny *datetime*, pomocí které je získáván aktuální datum. Teď, když už jsou stažená data z url stránky, využijeme knihovnu *BeautifulSoup*, která převede binární kód na html kód, který bude zpracováván. Tato knihovna umí třídit jednotlivé html třídy a vyčítat z nich data. Jelikož jsou spotové ceny i s několika dalšími daty uvedené na webu v tabulce, je potřeba najít v html datech třídu *table* a zároveň *report_table*, protože bylo zjištěno, že naše požadovaná data se ukrývají zároveň i pod touto třídou. Následně jsou data pro každý řádek rozdělena

do buněk, a poté z jednotlivých buněk vložena do proměnných pro každý typ dat. Pro tuto práci byly ale použity jen hodiny a spotové ceny, které jsou poté z funkce vráceny.

4.4.2 Stahování predikce osvětlení

Pro stahování predikce slunečního osvětlení je využito služby PV Forecast, jak už bylo zmíněno. Pro získání dat je zapotřebí sestavení dotazu s unikátním klíčem a polohou. Tyto hodnoty jsou zadány při registraci na webu. Další parametry dotazu, které je možné nastavovat jsou formát dat, typ sumy, délka předpovědi, automatická volba časové zóny nebo také začátek předpovědi. Detailní popis získávání dat je možné přejít si v citované dokumentaci. [40]

Python kód opět využívá knihovnu *requests* pro získání dat pomocí http požadavku. Tato data jsou tentokrát převedena na json formát. Výstup těchto dat mají podobu hodnoty predikce osvětlení přiřazené ke každé hodině dne. V jednoduchém for cyklu je pak počítána výroba fotovoltaické elektrárny pro každou hodinu pomocí rovnice 1 a do další proměnné jsou tyto hodnoty hodinové výroby přičítány pro celkovou denní výrobu. Následně jsou z funkce tato data vrácena. Zde je zdrojový kód pro tuto funkci:

```
def fetch_and_calculate():
    try:
        response = requests.get(API_URL)
        response.raise_for_status() # Ověří, zda server nevrátil chybu
        data = response.json() # Převede odpověď na JSON
        vykon_za_hodinu = [] # Pole pro výkon v každé hodině
        hodiny = []
        celkovy_vykon = 0

        for cas, osvit in data: # Projde každou hodnotu osvětlení
            P = osvit * A * ucinnost * pocet_panelu / 1000
            if P > 7.7: #omezení max výroby FVE
                P = 7.7
            hodiny.append(cas)
            vykon_za_hodinu.append(P) # Uložení do pole
            celkovy_vykon += P
        return vykon_za_hodinu, hodiny, celkovy_vykon

    except requests.exceptions.RequestException as e:
        print(f"Chyba při stahování dat: {e}")
        return None # Pokud nastane chyba, vrátí None
```

4.4.3 Komunikace mikropočítače se střídačem

Tuto komunikaci zajišťuje protokol Modbus TCP. Zvolení tohoto protokolu bylo dáno především tím, že jej podporuje střídač. Dalším důvodem je skutečnost, že v některých případech je potřeba číst více registrů najednou, a komunikace přes Modbus RTU by byla v takových situacích příliš pomalá. Obě zařízení jsou navíc připojena do stejné sítě, takže nevznikl žádný další problém s konektivitou. Modbus TCP je v tomto případě rychlejší a více vhodný než Modbus RTU.

Pro vytvoření funkcí pro čtení různých registrů ze střídače je zapotřebí knihovna *pymodbus*. Konkrétně třídu z *pymodbus.client* s názvem *ModbusTcpClient*. Pro připojení je nutno znát IP adresu střídače, která je v lokální síti rezervována na *192.168.0.226*, a jeho port, který je standardně nastaven na *502*. Následuje připojení ke střídači pomocí importované třídy.

```
# Připojení k střídači
def connect_client():
    # Vytvoření Modbus TCP klienta
    client = ModbusTcpClient(stredic_ip, port=stredic_port, timeout=5)
    #pokud se nepodaří připojit ke klientovi (střídači) vypíše hlášku
    if not client.connect():
        print("Nelze se připojit ke střídači!")
        return None
    return client
```

Pokud se tedy připojení ke střídači podaří, může následovat vyčítání registrů. Seznam registrů je dostupný na webu prodejce střídače. [41] Pro výpočet denní spotřeby je potřeba vyčíst pět registrů. Všechny tyto registry obsahují výkon v aktuální den, proto je nutné je číst těsně před koncem dne, tedy těsně před půlnocí. Nachází se v *input_registers* (s funkčním kódem 0x04) a jedná se o vyrobenou energii (0x0096), dodanou energii do sítě (0x0098), odebranou energii ze sítě (0x009A), množství vybité energie z akumulátoru (0x0020) a množství nabití energie do akumulátoru (0x0023). Z těchto dat je pak vypočítána denní spotřeba pomocí rovnice 2.

$$\text{denní spotřeba} = \text{vyrobena} + \text{odebráno} - \text{dodáno} + \text{nabito} - \text{vybito} \quad (2)$$

Dále je vyčítán registr, který obsahuje aktuální kapacitu akumulátoru (0x001C) pro výpočet spotřeby na další den, tedy kolik energie bude další den potřeba dobít. Tato hodnota se čte taktéž jednou denně, v době, kdy se rozhoduje o toku energie pro další den. Další registr je s hodnotou teploty akumulátoru (0x0018), který slouží čistě pro vizualizaci, jelikož si tuto hodnotu vyčítá samotné PLC, které sepne topné těleso u akumulátoru, pokud klesne teplota pod danou hodnotu, aby se zamezilo poškození akumulátoru. Tento údaj je mikropočítačem čten jednou za minutu. Poslední vyčítaný registr je dodávaný/odebíraný výkon ze sítě (0x0046). Tento registr je střídačem aktualizován v reálném čase, proto je taky mikropočítačem čten každé dvě sekundy. Z těchto hodnot lze určit, zda systém funguje správně a není například v době, kdy je nadbytek energie, odebírána energie ze sítě z důvodu připojení vysoké přídavné zátěže. Pro řízení si tuto hodnotu vyčítá PLC a spíná podle ní přídavnou zátěž.

Pro zakazování nabíjení akumulátoru je potřeba také funkce pro zápis do registru. Jde o nastavování maximálního nabíjecího proudu (0x0024). Jedná se o zápis a funkční kód je tedy 0x06. Se stejným principem je také zapisován zákaz exportu do sítě do registru uživatelského limitu pro export (0x0042). Nastavování maximálního proudu bylo zvoleno z důvodu pouze dvou zápisů do registru při zakázání a povolení nabíjení. Další možností je měnit mód střídače ze *self-use* na *manual* a poté zakázat nabíjení akumulátoru. Zde by šlo o tři zápisy pro jednu hodinu. Z důvodu životnosti střídače, který si zápisy do registru trvale ukládá do paměti EEPROM, která má omezený počet zápisů, je doporučeno jen několik jednotek zápisů denně. Proto i výsledné zefektivnění toku energie bude pouze pro maximálně čast zápisy denně kvůli tomuto omezení. Příklad zdrojového kódu funkce pro čtení registru:

```
def read_battery_capacity(client):
    try:
        response8 = client.read_input_registers(28,1)
        time.sleep(0.2)
        if not response8.isError():
            bat_capacity = response8.registers[0] #jednotka 1%
            print(f"Kapacita baterky: {bat_capacity}")
            return bat_capacity
        else:
            print(f"Chyba při čtení battery capacity: {response8}")
    except Exception as e:
        print("Chyba:", e)
```

4.4.4 Komunikace mikropočítače s PLC

Mikropočítač byl připojen ke sběrnici RS485, na které již probíhá komunikace PLC se střídačem pomocí protokolu Modbus RTU. Při tomto protokolu může být na sběrnici jedno zařízení tzv. master a zbytek zařízení jsou tzv. slave. Master ovládá komunikaci na sběrnici a slave zařízení jenom odpovídají. Před přidáním mikropočítačové jednotky bylo master zařízení PLC, a tak by to mělo i zůstat, aby při případném výpadku mikropočítače dál fungovalo alespoň spínání přídavné zátěže. Zavedení spojení mezi PLC a mikropočítačem je z důvodu ověření funkčnosti topení akumulátoru. Tedy v následné vizualizaci není dostačující pouze teplota akumulátoru, ale je potřeba také stav topení akumulátoru. Ke komunikaci přes sériovou sběrnici jsou využity dva piny UART, jejichž úlohou je vyslat datové slovo, získané od procesoru, bit po bitu do sériové sběrnice. Taktéž slouží pro složení datového slova načtených ze sériové sběrnice. Jelikož je RS485 poloduplexní, je nutné přepínat směr linkového budiče do režimu přijímání či odesílání. Přepínání režimů je přivedeno na GPIO pin 18 a pomocí *lgpio* knihovny jsou režimy ovládány. [42]

Pro Modbus komunikaci na sběrnici ze strany mikropočítače byla použita knihovna *pymodbus*, tentokrát ale její část pro rtu komunikaci. Jelikož je master zařízení PLC, je nutné v mikropočítači připravit paměť pro simulaci registrů, do kterých bude PLC data zapisovat.

```
store = ModbusSlaveContext(  
    di=ModbusSequentialDataBlock(0, [0] * NUM_REGISTERS),  
    co=ModbusSequentialDataBlock(0, [0] * NUM_REGISTERS),  
    hr=ModbusSequentialDataBlock(0, [0] * NUM_REGISTERS),  
    ir=ModbusSequentialDataBlock(0, [0] * NUM_REGISTERS)  
)
```

Další potřebná část pro komunikaci je spuštění serveru, který definuje potřebné parametry, pro přenos zpráv po sběrnici a zároveň aby případné odpovědi měly podobu, které dokáže zpracovat PLC. Tyto parametry musí mít stejné všechny zařízení na sběrnici, tedy baudrate 19200, bytesize 8, stopbits 1 a parity N. Tento server musí běžet po celou dobu komunikace, proto běží na samostatném vlákně na pozadí programu. Toho bylo docíleno pomocí knihovny *threading*.

Pro přečtení dat zapsaných do simulovaných registrů z PLC, byla vytvořena funkce. V této funkci je čten pouze vyšší byte konkrétního registru, do kterého by měla být zapsána data z PLC.

```
def monitor_registers():
    try:
        # Přečteme hodnoty z paměti registrů
        registers = context[0].getValues(3, REGISTER_ADDRESS,
COUNT_REGISTERS)
        bat_on_off = registers[0] >> 8 & 0xFF
        return bat_on_off
    except Exception as e:
        print(f"Chyba při čtení registrů: {e}")
```

4.4.5 Databáze

Pro ukládání dat a následnou jejich vizualizaci byla vybrána InfluxDB. Disponuje verzí zdarma a je vhodná pro zápis i čtení v reálném čase což naše aplikace vyžaduje. Po založení databáze na webu je vytvořen unikátní token, který je potřeba k připojení k databázi. Dále byl přes webové rozhraní vytvořen tzv. bucket, který byl pojmenován *Data* a organizace, která byla pojmenována jako *Home*. V programu se pak využívá knihovna *influxdb_client*. Pro připojení k databázi pro zápis pak slouží tato část kódu.

```
client = InfluxDBClient(url=url, token=token, org=org)
write_api = client.write_api(write_options=SYNCHRONOUS)
```

Následně jsou vytvořeny funkce pro zápis jednotlivých dat. Data v databázi se rozlišují díky tzv. Tagům, díky kterým víme o jaká data jde, když je potřebujeme číst. Ukázka funkce pro zápis do databáze:

```
def zapis_feeding_power(value):
    point = Point("feeding_power") \
        .tag("zdroj", "stridac") \
        .field("feeding_power", value)
    write_api.write(bucket=bucket, org=org, record=point)
```

V tomto programu byla také vytvořena univerzální funkce pro čtení dat z databáze. Tato funkce vyžaduje jaká data mají být přečtena, časový vzorek, který má být přečten, například za posledních několik dní a také do ní byl přidán filtr pro čtení podle dne v týdnu, například pouze pondělky. Tento filtr je přidán z důvodu výpočtu průměrné spotřeby pro

daný den, tzn. že požadují z databáze vyčíst spotřebu za posledních 5 pondělků, které se zprůměrují a vznikne tak průměrná spotřeba pro pondělí. Zde je zdrojový kód funkce pro čtení dat z databáze.

```
def nacti_data(measurement, minutes, weekday_filter = None):
    query = f'''
    from(bucket: "{bucket}")
      |> range(start: -{minutes}m)
      |> filter(fn: (r) => r._measurement == "{measurement}")
    '''
    if weekday_filter:
        query += f'''
        |> filter(fn: (r) => r["weekday"] == "{weekday_filter}")
        '''
    query_api = client.query_api()
    result = query_api.query(query, org=org)

    data = {}
    for table in result:
        for record in table.records:
            pole = data.setdefault(record.get_field(), [])
            pole.append(record.get_value())
    print(json.dumps(data))
```

4.4.6 Hlavní program

Využitím všech popsaných programů mohl vzniknout hlavní program. Vzhledem k tomu, že Raspberry Pi využívá vlastní repositář s knihovnamí, který neobsahuje knihovnu pro přístup do InfluxDB, a zároveň nelze kvůli systémové ochraně použít standardní příkaz *pip install*, bylo nutné zvolit alternativní postup. Jedinou možností, jak knihovnu nainstalovat a používat, je vytvoření a využití virtuálního prostředí. Vzhledem k tomu, že zbytek programu funguje přímo na mikropočítači, nikoliv ve virtuálním prostředí mikropočítače, bylo nutné vymyslet kompatibilitu. Ta se docílila spouštěním funkcí pro přístup do databáze ve virtuálním prostředí pomocí knihovny *subprocess*. Pomocí této knihovny je program spouštěn přes terminál se zapnutým virtuálním prostředím a výsledky funkcí jsou vráceny čtením výstupu v terminálu, tedy výsledky funkce *print*. Ve virtuálním prostředí se tedy spustí jen požadovaná funkce pro přístup do databáze a zbytek programu může fungovat

přímo na mikropočítači. Program nejprve spustí server databáze, a následně je volána požadovaná funkce přes cestu ve virtuálním prostředí.

```
subprocess.run(['sudo', 'systemctl', 'start', 'influxdb'])
python_path = os.path.expanduser('~/.myenv/bin/python')

subprocess.run([
    python_path, '-c',
    f"import influxDB; influxDB.zapis_spotreba({spotreba:0.2f})"
], check=True)
```

Hlavní program tedy na začátku definuje potřebné proměnné pro časové podmínky, jako jsou doby posledního přečtení teploty akumulátoru, denních statistik, predikce osvitů a také poslední hodinu, která slouží pro zápis do registru pouze jednou za hodinu. Všechny tyto proměnné, kromě poslední hodiny zápisu, jsou definované pomocí knihovny *datetime*. Na začátku programu je také spuštěn, již zmíněný, server databáze.

```
# Inicializace časových značek
last_bat_temp = datetime.min
last_daily = datetime.min
last_daily_osvit = datetime.min
last_hour = 0
```

Zbytek hlavního programu běží v nekonečné smyčce. Nekonečná smyčka se vykonává cyklicky, a to s dvousekundovým intervalem. Na začátku smyčky je volána funkce pro připojení ke střídači a na konci jeho odpojení. Odpojování od střídače je zásadní kvůli tzv. timeoutu, který běží, pokud je nějaké zařízení připojené ke střídači. Pokud by bylo zařízení připojené moc dlouho, timeout by vypršel a střídač odpojí připojené zařízení od TCP komunikace. Po připojení ke střídači je tedy čten dodávaný/odebíraný výkon ze sítě a ukládán do databáze. Poté následují časové podmínky pro čtení dalších registrů. Časová podmínka pro čtení teploty akumulátoru přes Modbus TCP a stav topení akumulátoru přes Modbus RTU, je jedna minuta. Přečtené hodnoty se hned vzápětí ukládají do databáze.

Další časová podmínka je pro čas 23:55, kdy jsou přečteny denní statistiky ze střídače. Tento čas je zvolen, kvůli resetování denních statistik o půlnoci. Do podmínky je přidáno porovnání uplynulých sekund, aby se zajistilo, že se akce neprovede vícekrát během jednoho časového okna. Všechny tyto statistiky, společně i z nich vypočítanou denní spotřebou, jsou taktéž vzápětí uloženy do databáze.

Nejdůležitější část celého programu, kde se mimo jiné řeší samotné řízení, je taktéž za časovou podmínkou. Tato část programu se provádí v čase 0:15, kdy jsou získána poslední data potřebná k řízení – a to predikce osvitu a spotové ceny. Tato data jsou stažena a uložena pro vizualizaci do databáze. Pro získání přebytečného výkonu, který je potřeba rozdělit na prodej pro nejvyšší spotové ceny, je zapotřebí získat ještě několik dat. Mezi ně patří výpočet energie, která je potřeba další den dobít do akumulátoru. K tomu je ze střídače získána aktuální úroveň nabití akumulátoru v procentech a následně je výpočet proveden podle rovnice 3, kde 11,6 představuje maximální množství elektrické energie v kWh, které je možné uložit do akumulátoru. Pokud z nějakého důvodu, nepřijdou data ohledně aktuální úrovně nabití akumulátoru, je tato hodnota nastavena na 80 %, aby program mohl dále pokračovat. Hodnota byla zvolena jako kompromis, aby nebylo výsledné řízení výrazně ovlivněno.

$$\text{potřebné dobít} = \left(1 - \frac{\text{aktuální stav nabití v procentech}}{100}\right) * 11,6 \quad (3)$$

Další potřebná data pro výpočet přebytečné energie je průměrná spotřeba pro nadcházející den. Proto jsou jednotlivé denní spotřeby ukládány také společně s číslem dne v týdnu reprezentující pondělí až neděli. Průměrná spotřeba je tedy získávána z databáze za 4 poslední týdny, s týdenním filtrem, podle dne v týdnu. Výsledkem tohoto čtení z databáze budou tedy 4 hodnoty spotřeby, například pro pondělí. Tyto hodnoty jsou pak zprůměrovány pro získání průměrné spotřeby pro daný den.

```
result6 = subprocess.run([
    python_path, '-c',
    f"import influxDB; influxDB.nacti_data(\"spotreba\", 40320, {weekday}) ",
    ], capture_output=True, text=True, check=True)

#data jsou ve formátu json aby se dal použít slovník z výstupu funkce
data = json.loads(result6.stdout)

#Vypocet prumerne spotreby
for key, values in data.items():
    if key == "spotreba":
        prumer_spotreba = sum(values) / len(values) if values else 0
```

Ted' už jsou dostupné všechna data potřebná k výpočtu přebytečné energie. Ta je získána odečtením průměrné spotřeby pro aktuální den v týdnu, energie potřebné k dobití akumulátoru a bezpečnostní konstanty od predikce denní výroby fotovoltaické elektrárny. Bezpečnostní konstanta je ve výpočtu z důvodu zohlednění nejistoty předpovědi a vytváří rezervu pro případ nepřesností v datech.

Pro zákaz exportu energie do sítě, či nabíjení akumulátoru byly vytvořeny seznamy dvaceti čtyř hodnot, které představují jednotlivé zákazy pro každou hodinu dne. Zákaz exportu energie do sítě má nastat, pokud spotové ceny klesnou pod nulu. Proto program přečte všechny stažené ceny pro aktuální den a pokud se v nich nachází záporná cena, zapíše pro odpovídající hodinu nulu do seznamu pro zákaz exportu energie. Tato nula představuje limit výkonu, který může být exportován do sítě.

Následuje logika pro zákaz nabíjení akumulátoru a spínání přídavné zátěže, tedy prioritizaci exportu. Zde je použit výpočet přebytečné energie, která, pokud je větší jak nula, se rozdělí do míst s nejvyšší spotovou cenou. Program tedy cyklicky začne procházet seznam s jednotlivými spotovými cenami a hodinami, který je seřazen od nevyšší spotové ceny po nejnižší. Následuje podmínka, která zjišťuje, jestli v tuto hodinu fotovoltaická elektrárna vyrábí. Zde se využije stažené hodinové predikce osvitu. Poté program zkontroluje, jestli má vůbec nějakou energii pro rozdělení a pokud už nepřekročil denní maximální počet zápisů do registru. Pokud je tato podmínka taktéž splněna, je od přebytečné energie odečtena předpokládaná výroba elektrárny pro danou hodinu, která je vypočítána z hodinové predikce osvitu, a do seznamu pro zákaz nabíjení akumulátoru je zapsána nula pro odpovídající hodinu. Zákaz nabíjení akumulátoru je pomocí registru maximálního proudu nabíjení, proto je zapsána nula.

U zákazu exportu energie do sítě i nabíjení akumulátoru bylo potřeba vymyslet limit počtu zápisů do registru, které si ukládá vestavěná paměť EEPROM s omezeným počtem zápisů. Tento limit byl stanoven na šest zápisů denně. Ať už se jedná o zákaz exportu do sítě nebo zákaz nabíjení akumulátoru. Po skončení jejich zákazu je nutné opětovné povolení a každý takový zásah tedy vyžaduje dva zápisy do registru (zákaz a následné povolení). Z tohoto důvodu byl v programu stanoven limit na maximálně tři takové zápisy. Při vytváření zákazů pro odpovídající hodiny je tedy do podmínek přidána kontrola maximálního počtu zápisů. Pokud ale nastane situace, kdy je tři hodiny po sobě zakázán například export energie do sítě, jedná se pouze o 2 zápisy, zakázání a povolení, tedy v programu o jeden zápis. Z tohoto důvodu byla přidána kontrola, která ověřuje, zda zákaz

nebyl již zapsán pro hodinu před nebo po nastavované hodině. Pokud bude podmínka vyhodnocena jako pravdivá, přičte se do počtu zápisů jednička. Pokud podmínka není splněna, tedy pokud již byl zákaz nastaven pro hodinu před nebo po nastavované hodině, nedojde k přičtení do počtu zápisů, protože se stávající zákaz pouze prodlouží o další hodinu.

Poslední část hlavního programu jsou samotné zápisy do registrů pro zákaz exportu do sítě a nabíjení akumulátoru. Tato část programu kontroluje každou hodinu dne, jestli má být zapsán zákaz či povolení. Zároveň je také kontrolována předchozí hodina, aby se stejný zápis neprováděl vícekrát po sobě.

```
for i in range(len(bat_on_off)):
    if i == hour_now and hour_now != last_hour:
        last_hour = hour_now
        if bat_on_off[i] == BAT_OFF and bat_on_off[i-1] != BAT_OFF:
            modbus_tcp.write_on_off_battery_charge(client, BAT_OFF)

        elif bat_on_off[i] == BAT_ON and bat_on_off[i-1] != BAT_ON:
            modbus_tcp.write_on_off_battery_charge(client, BAT_ON)

        if export_on_off[i] == EXPORT_OFF and export_on_off[i-1] !=
EXPORT_OFF:
            modbus_tcp.write_export_on_off(client, EXPORT_OFF)

        elif export_on_off[i] == EXPORT_ON and export_on_off[i-1] !=
EXPORT_ON:
            modbus_tcp.write_export_on_off(client, EXPORT_ON)
```

4.5 Úprava programu PLC

Hlavní úpravou je přidání čtení dalšího registru ze střídače. Kromě vyčítání přetoku do sítě (0x0046) pro řízení spínání přídatné zátěže a teploty akumulátoru (0x0018) pro spínání topného tělesa akumulátoru, bylo přidáno čtení limitu nabíjení akumulátoru (0x0090). Tento registr je řízen mikropočítačem a v PLC je jeho hodnota čtena pro rozhodnutí, zda má být přídatná zátěž zakázána. Tedy pokud je limit nabíjení akumulátoru nastaven na nulu neprovede se čtení přetoku do sítě, a do regulačního funkčního bloku je poslána hodnota - 1000, reprezentující 1 kW odběr ze sítě. Tato hodnota byla zvolena z důvodu postupného vypnutí všech přídatných zátěží. Tím se docílí priority exportu energie do sítě. [43]

```
IF BatCharging = 300 THEN
  RSSend_1_FeedingPower(Request:=
true,Adr:=0,Channel:=0,Start:=uint#100,Count:=uint#13,Answer:=true,A
sString:=false,AnswAdr:=uint#150,EndChar:=false,EC_Count:=uint#10,Ti
meOut:=t#200ms);

  IF RSSend_1_FeedingPower.Received THEN
    Wvykon_roled:=ROL(Wvykon,8);
    Rvykon:=WORD_TO_REAL(Wvykon_roled);
    BoilerRegulation_1(Rvykon:=Rvykon);
    Display_1(RVYkon:=Rvykon);
  END_IF;
ELSE
  BoilerRegulation_1(Rvykon:=-1000.0);
END_IF;
```

Další úpravou PLC programu je zasílání výstupu topného tělesa akumulátoru do mikropočítače. Tahle úprava slouží čistě pro vizualizaci a ověření, zda funguje spínání topného tělesa při nízké teplotě akumulátoru. Toho je docíleno zápisem do simulovaného registru v mikropočítači pomocí protokolu Modbus RTU.

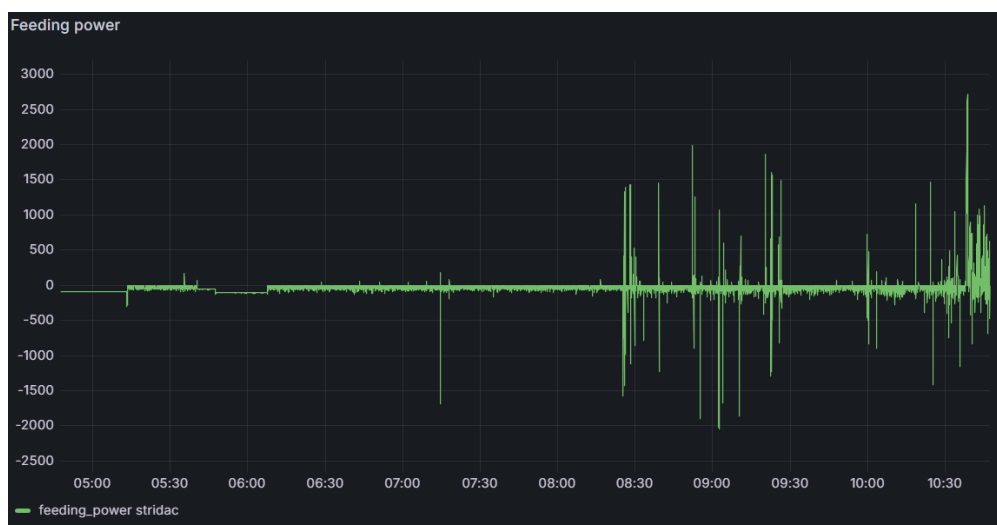
Vzhledem k velkému počtu operací probíhajících na sběrnici byl navržen stavový automat, který předchází jejich kolizím. Protože některé z těchto operací mohou trvat až několik sekund, byla četnost čtení teploty akumulátoru, limitu nabíjení a zápisu stavu topného tělesa omezena na jednu za minutu. Toto řešení minimalizuje zatížení sběrnice a zároveň neovlivňuje negativně rychlost regulace přídatné zátěže, u které je akceptovatelná odchylka v přesnosti jednou za minutu.

4.6 Vizualizace

Pro vizualizaci dat byla použita open-source platforma Grafana, která umožňuje přehledné zobrazení dat v reálném čase formou interaktivních grafů, tabulek a dalších prvků. Mezi hlavní výhody Grafany patří její široká podpora datových zdrojů, v tomto případě byla využita databáze InfluxDB, se kterou Grafana bez problémů spolupracuje. V Grafaně byly vytvořeny tři tzv. dashboardy, do kterých se přidávají jednotlivé panely, kterým je možné nastavovat veškeré parametry od výběru dat až po vzhled. V této práci byla vizualizována tato data:

- Dashboard Feeding power
 - Aktuální odběr/dodávka do sítě
 - Teplota akumulátoru
 - Stav topného tělesa akumulátoru
- Dashboard Denní statistiky
 - Denní statistiky
 - Spotřeba
- Dashboard Spot a osvit
 - Spotové ceny
 - Hodinová predikce výroby
 - Celková predikce výroby

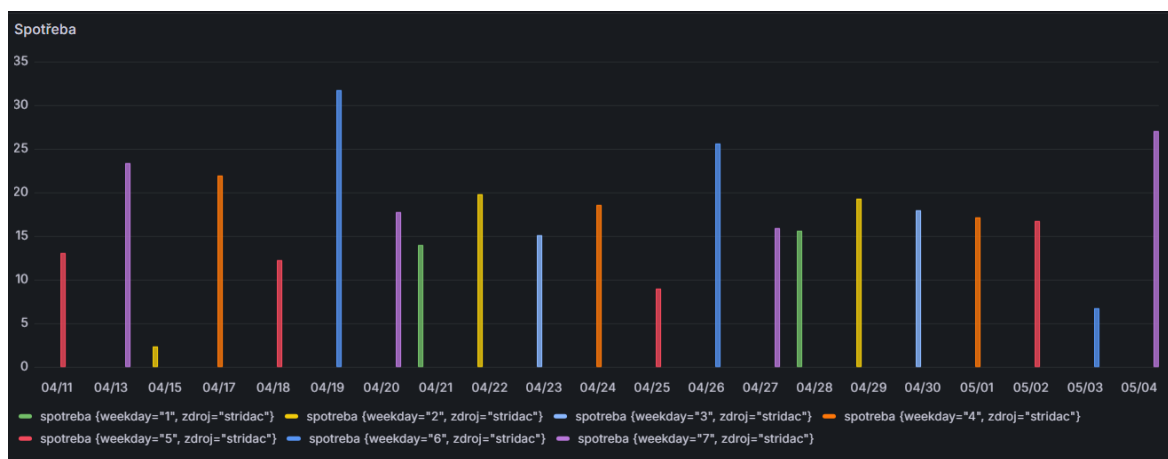
Rozdělení do těchto třech dashboardů má jediný důvod. Časový interval, který je požadován pro zobrazení dat, nelze určit pro každý panel zvlášť, ale pouze pro celý dashboard. Proto nemohou být denní statistiky, které jsou zobrazovány za několik posledních dní na stejném dashboardu jako spotové ceny a osvit, které jsou zobrazovány za poslední den nebo aktuální odběr/dodávka do sítě. Zároveň mají grafy omezení pro počet zobrazených bodů, takže právě graf s odběrem/dodávkou do sítě může být zobrazen maximálně pro posledních šest hodin tak, aby tento limit nebyl překročen.



Obrázek 14 Graf dodávky/odběru energie ze sítě ve vizualizaci Grafana

Denní statistiky			
datum	Odběr [kWh]	Přetok [kWh]	Výroba [kWh]
2025-04-28	1.98	1.91	14.3
2025-04-29	2.37	28.0	45
2025-04-30	3.38	7.76	22.4
2025-05-01	2.19	2.96	15.5
2025-05-02	1.48	3.20	17.9
2025-05-03	3.16	2.63	12.4
2025-05-04	1.77	1.55	18

Obrázek 16 Tabulka denních statistik ve vizualizaci Grafana

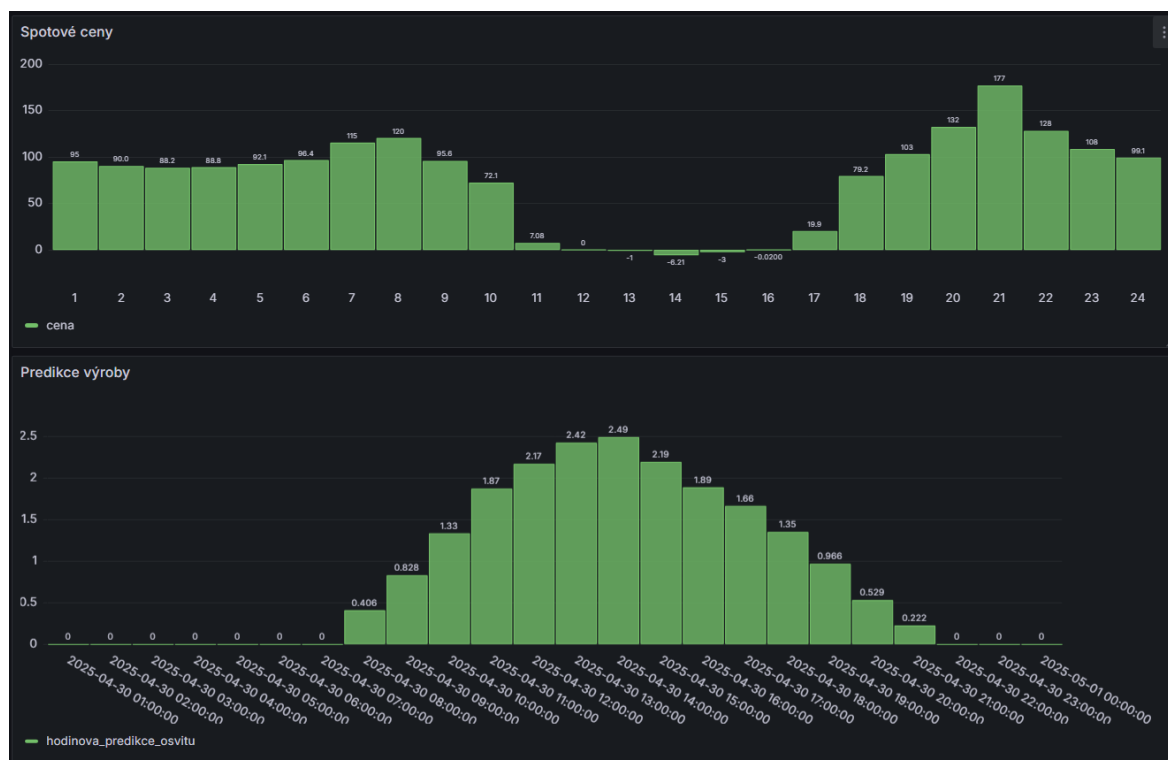


Obrázek 17 Graf denních spotřeb ve vizualizaci Grafana

5 OVĚŘENÍ FUNKČNOSTI JEDNOTKY

Ověření bylo provedeno pomocí Python funkce *print*, která vypisovala jednotlivé úkony programu, jako jsou přepínání GPIO pinu, spouštění serveru pro Modbus RTU komunikaci či zahájení stahování spotových cen, ukládání všech dat do databáze nebo také chybová hlášení. Zároveň jsou jednou denně při nastavování řízení také vypisovány jednotlivé mezi výpočty a nastavovací podmínky, aby mohlo být zkontrolováno, že program dané řízení navrhl správně. Většina těchto dat lze ověřit pouze ve vývojovém prostředí, a sloužily pro kontrolu funkčnosti programu. Jakmile byla ověřena funkčnost programu bylo nutné ověřit, zda se výstup programu opravdu projevuje na reálném systému. K tomu bylo využito, kromě vlastní vizualizace, vizualizace od výrobce střídače, která je po přihlášení dostupná na stránce <https://global.solaxcloud.com/>. Tahle vizualizace od výrobce aktualizuje data pouze jednou za pět minut, ale je zde hezky vidět celodenní porovnání grafů výroby, spotřeby či nabíjení akumulátoru.

Zde je popsána funkčnost jednotky pro den 30.4.2025, ve kterém se nacházely hodiny se zápornými spotovými cenami a zároveň předpoklad denní výroby pro tento den byl taktéž vysoký, pro ověření prioritizace exportu.



Obrázek 18 Graf spotových cen a predikce výroby

Z obrázku 18, konkrétně grafu spotových cen, lze vidět, že několik hodin tohoto dne jsou spotové ceny záporné. V této době je požadován nulový export, aby domácnost neplatila za dodávku energie do sítě. Na obrázku 19 na grafu zákazu exportu do sítě, lze vidět, že přesně v hodinách, kdy jsou záporné spotové ceny, je nastaven limit exportu do sítě na nulu. Tento limit znázorňuje maximální povolený výkon pro exportování energie do sítě, a pokud je tedy nastaven na nulu, je nastaven zákaz exportu energie do sítě.

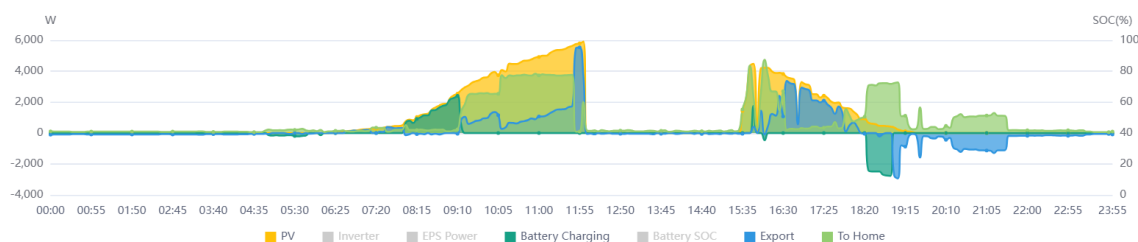


Obrázek 19 Graf zákazu exportu do sítě a nabíjení akumulátoru

Jelikož celková predikce výroby pro tento den byla 20,3 kWh, bylo potřeba taktéž rozdělit přebytečnou energii do hodin s nejvyšší spotovou cenou. Průměrná spotřeba a energie potřebná pro dobíjení akumulátoru byly v tento den 18,25 kWh. K této hodnotě je ještě přičtena bezpečnostní konstanta 1 kWh, a tedy výsledná přebytečná energie je rovna okolo 1 kWh. Tuto přebytečnou energii chceme prodat za co nejvyšší spotové ceny. Z porovnání obrázku 18 a 19 lze zřetelně poznat, že program vybral opravdu tři nejvyšší spotové ceny. Detailní mezi kroky a mezi výpočty, kdy program rozhoduje o zákaze exportu do sítě či nabíjení akumulátoru, lze také zkontrolovat ve vývojovém prostředí jako výstupy funkce

print. Další nutnost, která musí být ověřena je počet zápisů do registru střídače, který je programově omezen na maximálně šest zápisů denně, jak už bylo dříve popsáno. Jak lze vidět na obrázku 19, tak v tento den se jednalo přesně o šest zápisů, a to dvakrát vypnutí a zapnutí nabíjení akumulátoru a jednou vypnutí a zapnutí exportu energie do sítě. Taktéž bylo ve vývojovém prostředí ověřeno, že program neprovádí do registru zápis, pokud se jedná o stejnou hodnotu, tedy zápis je prováděn pouze při změně.

Reálné chování pak lze pozorovat na obrázku 20, který je z vizualizace od výrobce. Graf znázorňuje výrobu žlutou barvou, spotřebu zelenou barvou, odběr či export do sítě modrou barvou a nabíjení či vybíjení akumulátoru tmavě zelenou barvou. Obrázek 20 si lze také pohlédnout v příloze I, kde je zobrazen na šířku. Zároveň do přílohy II byl přidán graf jiného dne, kde lze lépe vidět odložení nabíjení akumulátoru.



Obrázek 20 Graf chodu systému

Na obrázku 20 je patrné, že nabíjení akumulátoru bylo zahájeno až v osm hodin, což odpovídá plánovanému odložení zobrazenému na obrázku 19. Následně nastanou ve dvanáct hodin záporné spotové ceny, kde lze velmi pěkně vidět, že opravdu od dvanácti do šestnácti hodin nebyla exportována žádná energie z domácnosti. Od devatenácti hodin, kdy podle předpovědi výroby, by měla elektrárna vyrobit ještě necelou kWh, systém vypnul v tuto dobu nabíjení akumulátoru a spínání přídatné zátěže. V tuto dobu už však elektrárna nic nevyrábí, proto řízení v této době na systém nemá vliv.

ZÁVĚR

Cílem této bakalářské práce bylo navrhnout a implementovat systém pro sledování a řízení toku energie v domácnosti s fotovoltaickou elektrárnou. Důraz byl kladen na efektivní využití přebytků elektrické energie a zamezení nežádoucímu exportu do distribuční sítě. Motivací bylo snížení nákladů za elektrickou energii, potažmo prodeje přebytečné energie za co nejvyšší spotové ceny.

Pomocí mikropočítače Raspberry Pi byl realizován sběr dat ze střídače prostřednictvím Modbus komunikace. Součástí řešení byly i spotové ceny a predikce výroby, která byla vypočítána na základě dostupných dat o predikci slunečního osvitu z webu PV Forecast, doplněná bezpečnostní konstantou pro eliminaci možných odchylek. Pomocí těchto dat bylo navrženo řízení zákazu exportu přetoků do sítě a vypnutí nabíjení akumulátoru dle zvolených podmínek.

Pro uložení a další práci s časově závislými daty byla využita databáze InfluxDB, kam Raspberry Pi periodicky zapisuje informace o výrobě, spotřebě domácnosti, teplotě akumulátoru a dalších systémových veličinách. Tyto hodnoty byly následně vizualizovány pomocí platformy Grafana, což umožnilo vytvoření přehledných dashboardů pro analýzu chování systému v čase. Tato vizualizace byla důležitým nástrojem pro ladění i zpětné vyhodnocení regulační logiky.

Pro správné fungování musel být upraven i program PLC. Vzhledem k omezené propustnosti RS485 sběrnice byl implementován stavový automat, který řídí jednotlivé dotazy na sběrnici tak, aby nedocházelo ke kolizím. Vybrané dotazy byly optimalizovány na periodické spouštění s minutovým intervalem, aby nebylo zpomaleno řízení spínání přídatné zátěže.

V budoucnu je možné systém dále rozšířit například o přidání chytrého senzoru teploty bojleru a následné začlenění bojleru do řízení jako tepelného akumulátoru. Tím by bylo možné efektivně využívat přebytky energie k ohřevu vody podle aktuální potřeby a kapacity. Dále se nabízí rozšíření řízení i na stranu nákupu elektřiny podle spotových cen, zatímco současná verze systému optimalizuje pouze prodej přetoků podle těchto cen. Takové rozšíření by umožnilo ještě efektivnější hospodaření s energií v reálném čase a přiblížilo systém plnohodnotnému energetickému managementu domácnosti.

SEZNAM POUŽITÉ LITERATURY

- [1] *Fakta o klimatu*. Online. C2024. Dostupné z: <https://faktaoklimatu.cz/>. [cit. 2024-12-22].
- [2] *Jak se vyrábí elektrina v Česku?* Online. In: SLOVENSKÝ PLYNÁRENSKÝ PRIEMYSEL CZ [SPP CZ]. SPP. 2024. Dostupné z: <https://www.spp.cz/magazin/jak-se-vyrabi-elektrina-v-cesku>. [cit. 2024-12-23].
- [3] *Tepelná elektrárna*. Online. In: VIDA!. C2024. Dostupné z: <https://vida.cz/exponaty/tepelna-elektrarna>. [cit. 2024-12-23].
- [4] SLAVÍK, Ing. Jakub. *Víte, jak funguje jaderná elektrárna?* Online. In: Pro elektrotechniky. 2013. Dostupné z: <https://www.proelektrotechniky.cz/vzdelavani/12.php>. [cit. 2024-12-23].
- [5] *Jaderná elektrárna: Schéma nejběžnějšího typu jaderné elektrárny s tlakovodním reaktorem*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 29. 11. 2024. Dostupné z: https://cs.wikipedia.org/wiki/Jadern%C3%A1_elektr%C3%A1rna. [cit. 2024-12-23].
- [6] *Elektrárny na biomasu*. Online. In: Informační portál. 2017-2021. Dostupné z: <https://www.informacni-portal.cz/clanek/elektrarny-na-biomasu>. [cit. 2024-12-23].
- [7] VOBOŘIL, David. *Vodní elektrárny - princip, rozdělení, elektrárny v ČR*. Online. In: O energetice.cz. 2016. Dostupné z: <https://oenergetice.cz/obnovitelne-zdroje/vodni-elektrarny-princip-a-rozdeleni>. [cit. 2025-01-03].
- [8] *O vodní energetice*. Online. In: Skupina ČEZ. C2025. Dostupné z: <https://www.cez.cz/cs/o-cez/vyrobní-zdroje/obnovitelne-zdroje/voda/informace-o-vodni-energetice>. [cit. 2025-01-03].
- [9] *Přečerpávací vodní elektrárna Dlouhé stráně*. Online. In: Skupina ČEZ. C2025. Dostupné z: <https://www.cez.cz/cs/o-cez/vyrobní-zdroje/obnovitelne-zdroje/voda/vodni-elektrarny/ceska-republika/dlouhe-strane-58155>. [cit. 2025-01-03].
- [10] *Jak funguje větrná elektrárna*. Online. In: Skupina ČEZ. C2025. Dostupné z: <https://www.cez.cz/cs/o-cez/vyrobní-zdroje/obnovitelne-zdroje/vitr/jak-funguje-vetrna-elektrarna>. [cit. 2025-01-04].

- [11] VOBOŘIL, David. *Větrné elektrárny - princip, rozdělení, elektrárny v ČR*. Online. In: O energetice.cz. 2015. Dostupné z: <https://oenergetice.cz/elektrina/vetrne-elektrarny-princip-cinnosti-zakladni-rozdeleni>. [cit. 2025-01-04].
- [12] LÁZOKOVÁ, Eva. *Jak funguje fotovoltaická elektrárna*. Online. In: Woltair. 2024. Dostupné z: <https://www.woltair.cz/blog/fotovoltaika/jak-funguje-fotovoltaicka-elektrarna?pinnedQuestion=panelSize&onlyUnansweredQuestions=>. [cit. 2024-11-07].
- [13] *Fotoelektrický jev*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 22.4.2024. Dostupné z: https://cs.wikipedia.org/wiki/Fotoelektrick%C3%BD_jev. [cit. 2024-11-07].
- [14] HENZE, Andreas a HILLEBRAND, Werner. Komponenty solárních panelů. In: HENZE, Andreas a HILLEBRAND, Werner. *Elektrický proud ze slunce: fotovoltaika v praxi*. Ostrava: HEL, 2000, s. 29-54. ISBN 80-86167-12-7.
- [15] LÍSKOVEC, Pavel [@Loxone CZSK]. *Jak jsem zkrotil FVE a SPOT – Loxone Explained [4K] (2024)*. Online. 2024. Dostupné z: Youtube, https://www.youtube.com/watch?v=RFew0XOTEK4&ab_channel=LoxoneCZSK. [cit. 2025-01-04].
- [16] *Spotové ceny elektřiny a jak jim rozumět*. Online. In: Deltagreen. 2024. Dostupné z: <https://www.deltagreen.cz/blog/spotove-ceny-elekriny-a-jak-jim-rozumet>. [cit. 2024-11-07].
- [17] *Loxone Electronics*. Online. C2025. Dostupné z: <https://www.loxone.com/cscz/>. [cit. 2025-05-17].
- [18] *Huawei Technologies*. Online. C2025. Dostupné z: <https://solar.huawei.com/cz/>. [cit. 2025-05-17].
- [19] *Energy Flow Monitor*. Online. In: Loxone. C2025. Dostupné z: <https://www.loxone.com/enen/kb/energy-flow-monitor/>. [cit. 2025-05-21].
- [20] DOLINAY, Jan, ed., 2024. *Princip mikropočítače: Stručný přehled informací o fungování počítače (tedy i mikropočítače)*. [cit. 2024-11-21].
- [21] *Difference between MCU and SoC*. Online. In: Geeks for Geeks. 20.5.2024. Dostupné z: <https://www.geeksforgeeks.org/difference-between-mcu-and-soc/>. [cit. 2025-01-05].

- [22] BRADT, Paul a BRADT, David. *Science and engineering projects using the Arduino and Raspberry Pi: explore STEM concepts with microcomputers*. [United states]: Apress, 2020. ISBN 9781484258118.
- [23] *Raspberry Pi Hardware*. Online. In: Raspberry Pi. Dostupné z: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>. [cit. 2025-02-21].
- [24] *Raspberry Pi Zero*. Online. In: Canna Kit. C2025. Dostupné z: <https://www.canakit.com/raspberry-pi-zero.html>. [cit. 2025-02-21].
- [25] WHITE, Jane. *Introduction to Raspberry Pi 5: Specifications and Features*. Online. In: Designspark. 2024. Dostupné z: <https://www.rs-online.com/designspark/introduction-to-raspberry-pi-5-specifications-and-features>. [cit. 2024-11-04].
- [26] *Raspberry Pi 5 Single Board Computers*. Online. In: Mouser Electronics. 2023. Dostupné z: <https://cz.mouser.com/new/raspberry-pi/raspberry-pi-5-sbc/>. [cit. 2024-11-07].
- [27] *Python v ČR*. Online. Dostupné z: <https://python.cz/>. [cit. 2024-11-21].
- [28] HELLER, Martin. *What is Visual Studio Code? Microsoft's extensible code editor*. Online. In: InfoWorld. 8 Jul 2022. Dostupné z: <https://www.infoworld.com/article/2335960/what-is-visual-studio-code-microsofts-extensible-code-editor.html>. [cit. 2025-05-07].
- [29] *Remote Development using SSH*. Online. In: Visual Studio Code. 2025. Dostupné z: <https://code.visualstudio.com/docs/remote/ssh>. [cit. 2025-05-07].
- [30] PLC-automatizace.cz. *PLC – Programovatelný logický automat*. Online. Dostupné z: <http://plc-automatizace.cz/knihovna/plc.htm>. [cit. 2024-11-21].
- [31] *PLC - Programovatelný logický automat*. Online. In: Oneindustry. 2019. Dostupné z: <https://www.oneindustry.cz/lexikon/plc-programovatelný-logický-automat/>. [cit. 2024-11-24].
- [32] PINĎÁK, Bc. Jiří. *Systém kontroly a řízení železniční stanice Control and management system of Railway Station*. Diplomová práce. Zlín: UTB, 2024.

- [33] SYSALA, Tomáš. *PLC obecná přednáška*. Prezentace. Zlín, 2024. Dostupné také z: https://utbcz-my.sharepoint.com/:p:/r/personal/sysala_utb_cz/_layouts/15/Doc.aspx?sourcedoc=%7B4656268A-BF63-40B3-858AE640DB4AE914%7D&file=02%20PLC%20%C3%9Avodn%C3%AD%20obecn%C3%A1%20p%C5%99edn%C3%A1%20C5%A1ka%202024%20pro%20AP3PP.pptx&action=edit&mobileredirect=true.
- [34] POUCHA, Ing. Pavel. *Jak pracovat s linkou RS485 (Modbus RTU) nejen u FV elektráren*. Online. In: Papouch Store. C2024. Dostupné z: <https://papouch.com/jak-zapojit-rs485-modbus-rtu-fv-elektrarna/>. [cit. 2024-11-21].
- [35] *Odpojte rozdíl mezi Modbus RTU a Modbus TCP*. Online. In: Chilink IoT. Dostupné z: <https://www.chilinkiot.com/cs/unraveling-the-difference-between-modbus-rtu-and-modbus-tcp/>. [cit. 2025-04-24].
- [36] KW-SOFTWARE GMBH. *MULTIPROG Quick Start Guide*. Online. 2002. Dostupné z: <http://kgs.no/Produkter/Manual/Programvare/KV-Multiprog.pdf>. [cit. 2025-05-13].
- [37] *Fotovoltaický panel München Solar 550W, mono, half cut, stříbrný rám (paleta 31ks)*. Online. In: Eshop elektronika. Dostupné z: <https://www.eshopelektronika.cz/fotovoltaicky-panel-munchen-solar-550w-mono-half-cut-stribrny-ram-paleta-31ks>. [cit. 2025-04-22].
- [38] *Baterie Solax Triple Power T58 s výkonem 5,8kW*. Online. In: SVP Solar. Dostupné z: <https://www.solar-eshop.cz/p/baterie-solax-triple-power-t58-s-vykonem-5-8kw/varianta-baterie-master/>. [cit. 2025-04-22].
- [39] *Řada X3-Hybrid Uživatelská příručka: 5,0kW - 15,0kW*. Online. In: GBC Solino. C2018-2025. Dostupné z: https://gbc-solino.cz/wp-content/uploads/2023/09/MNL_SOLAX_X3_HYBRID_G4_5.0-15.0-CZ.pdf. [cit. 2025-02-07].
- [40] MARTÍNEK, RNDr. Vladislav. *PV Forecast WEB API: Popis webového rozhraní pro získání předpovědních dat ze služby PV Forecast*. Online. In: UNIVERZITNÍ CENTRUM ENERGETICKY EFEKTIVNÍCH BUDOV ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE [UCEEB, ČVUT]. PV Forecast. C2016. Dostupné z: https://wp2.pvforecast.cz/wp-content/uploads/2024/10/PV_Forecast_WEB_API_2.3.pdf. [cit. 2025-04-09].

- [41] *Energy Storage Inverter Modbus TCP&RTU Communication protocols*. Online. In: GBC Solino. 2020, 11.1.2022. Dostupné z: https://gbc-solino.cz/wp-content/uploads/2022/07/Hybrid-X1X3-G4-ModbusTCPRTU-V3.21-English_0622-public-version.pdf. [cit. 2025-04-09].
- [42] MANN, Burkhard. *C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky*,-. Praha: BEN, 2003. ISBN 80-730-0077-6.
- [43] *OMC8000: Základní modul PLC systému*. Online. In: Orbit Merret. 2016. Dostupné z: https://www.orbitmerret.eu/en/document-download?document_id=5008. [cit. 2025-05-17].

SEZNAM OBRÁZKŮ

Obrázek 1 Schéma nejběžnějšího typu jaderné elektrárny s tlakovodním reaktorem [5]....	16
Obrázek 2 Fotoelektrický jev [13]	22
Obrázek 3 Ukázka sledování toku energie od společnosti Loxone [19]	24
Obrázek 4 Raspberry Pi 5 [26].....	27
Obrázek 5 Otočka cyklu PLC [33].....	31
Obrázek 6 Vývojové prostředí Multiprog	32
Obrázek 7 Fotovoltaické panely na domě	35
Obrázek 8 Vývojový diagram ukazující práci chytrého střídače s energií	37
Obrázek 9 Vývojový diagram regulace výkonu bojleru	38
Obrázek 10 Program pro regulaci výkonu bojleru pomocí PID regulátoru	39
Obrázek 11 Program pro spínání přídavných topných těles	39
Obrázek 12 Blokový diagram systému, včetně implementovaného mikropočítače	42
Obrázek 13 Raspberry Pi a napěťový převodník pro připojení ke sběrnici umístěné v rozvaděči	44
Obrázek 14 Graf dodávky/odběru energie ze sítě ve vizualizaci Grafana	57
Obrázek 15 Grafy pro spotové ceny a predikci výroby ve vizualizaci Grafana	58
Obrázek 16 Tabulka denních statistik ve vizualizaci Grafana	59
Obrázek 17 Graf denních spotřeb ve vizualizaci Grafana	59
Obrázek 18 Graf spotových cen a predikce výroby	60
Obrázek 19 Graf zákazu exportu do sítě a nabíjení akumulátoru	61
Obrázek 20 Graf chodu systému	62

SEZNAM TABULEK

Tabulka 1 Porovnání vybraných modelů Raspberry Pi [23][22][24]	26
Tabulka 2 Parametry použitého Raspberry Pi 5 [23].....	28

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

°C	Stupeň Celsia – jednotka teploty
A	Ampér – jednotka elektrického proudu
ABS	Protiblokovací systém (Anti-lock Brake System)
CPU	Centrální procesní jednotka (Central processing unit)
CRC	Cyklická redundantní kontrola (Cyclic redundancy check)
EEPROM	Elektricky vymazatelná paměť pouze pro čtení (Electrically Erasable Programmable Read-Only Memory)
FVE	Fotovoltaická elektrárna
HDMI	Multimediální rozhraní s vysokým rozlišením (High-Definition Multimedia Interface)
ID	Identifikátor
kB	kilobyte – jednotka kapacity paměti
kW	kilowatt – jednotka výkonu
kWh	kilowatt hodina – jednotka elektrické energie
kWp	kilowatt peak – jednotka maximálního výkonu solárního panelu
mA	miliampér – jednotka elektrického proudu
MB	megabyte – jednotka kapacity paměti
MS	Microsoft – technologická společnost
MW	megawatt – jednotka výkonu
PLC	Programovatelný logický automat (Programmable logic controller)
PWM	Pulse width modulation (pulzně šířková modulace)
SSR	Solid state relay
UART	Universal Asynchronous Receiver and Transmitter (univerzální asynchronní přijímač a vysílač)
USB	Univerzální sériová sběrnice (Universal serial bus)
V	Volt – jednotka elektrického napětí

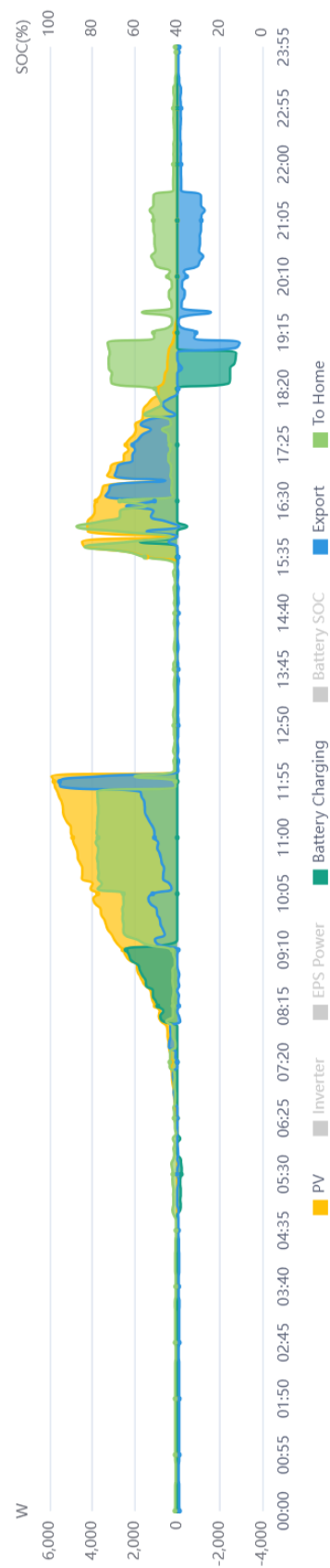
Wh	Watt hodina – jednotka elektrické energie
Wp	Watt peak – jednotka maximálního výkonu solárního panelu

SEZNAM PŘÍLOH

Příloha P I: Graf chodu systému

Příloha P II: Graf chodu systému při odkladu nabíjení akumulátoru

PŘÍLOHA P I: GRAF CHODU SYSTÉMU



**PŘÍLOHA P II: GRAF CHODU SYSTÉMU PŘI ODKLADU
NABÍJENÍ AKUMULÁTORU**

