

Návrh aplikace pro správu a řízení fotovoltaických elektráren

Vojtěch Janeček

Bakalářská práce
2024



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Vojtěch Janeček**
Osobní číslo: **A21702**
Studijní program: **B0613A140020 Softwarové inženýrství**
Forma studia: **Kombinovaná**
Téma práce: **Návrh aplikace pro správu a řízení fotovoltaických elektráren**
Téma práce anglicky: **Application Design for the Management and Control of Photovoltaic Power Plants**

Zásady pro vypracování

1. Popište současný stav řízení fotovoltaických elektráren případně velkokapacitních baterií.
2. Vysvětlete základní pojmy a nastiňte princip fungování aplikace.
3. Navrhněte model pro správu a řízení fotovoltaických elektráren a velkokapacitních baterií.
4. Navrhněte webovou SCADA aplikaci pro správu a řízení fotovoltaických elektráren a velkokapacitních baterií.
5. Popište standardy a požadavky na spolehlivost a bezpečnost vytvořené aplikace.
6. Nastiňte přínos navržené aplikace.

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

Seznam doporučené literatury:

1. ŠMEJKAL, Ladislav a MARTINÁSKOVÁ, Marie, 1999. PLC a automatizace. Praha: BEN – technická literatura. ISBN 80-860-5658-9.
2. PILGRIM, Mark, c2010. Ponořme se do Python(u) 3: Dive into Python 3. CZ.NIC. Praha: CZ.NIC. ISBN 978-80-904248-2-1.
3. MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3, 2012. Online. <https://modbus.org/>. Dostupné z: https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.
4. KAMENÍK, Pavel, 2011. Příkazový řádek v Linuxu: praktická řešení. Brno: Computer Press. ISBN 978-80-251-2819-0.

Vedoucí bakalářské práce: **doc. Ing. Bc. Bronislav Chramcov, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: **5. listopadu 2023**

Termín odevzdání bakalářské práce: **13. května 2024**



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

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 5. ledna 2024

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že 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;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že 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 samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.
- že při tvorbě této práce jsem použil nástroj generativního modelu AI ChatGPT; <https://chatgpt.com/> za účelem kvalitnější formulace textu. Po použití tohoto nástroje jsem provedl/a kontrolu obsahu a přebírám za něj plnou zodpovědnost

Ve Zlíně, dne

Vojtěch Janeček, v.r.
podpis studenta

ABSTRAKT

Cílem této bakalářské práce je vyvinout webovou SCADA aplikaci určenou pro efektivní správu a optimalizované řízení fotovoltaických elektráren. Tato aplikace bude primárně navržena tak, aby automaticky získávala informace o aktuálních cenách výkupu elektrické energie a následně automatizovala procesy řízení jak samotných fotovoltaických elektráren, tak i přidružených velkokapacitních bateriových úložišť. Klíčovým přínosem aplikace bude schopnost integrace a centralizace ovládání různých zařízení přes jednoduché webové rozhraní, které uživatelům poskytne přístup k reálným datům a umožní jim provádět vzdálené ovládání.

Klíčová slova: PLC automaty, SCADA (dispečerské řízení a sběr dat), fotovoltaické elektrárny, přenosové protokoly, Python (programovací jazyk)

ABSTRACT

The goal of this bachelor's thesis is to develop a web-based SCADA application designed for efficient management and optimized control of photovoltaic power plants. The application will primarily be designed to automatically gather information on current electricity buy-back prices and subsequently automate the control processes of both the photovoltaic plants themselves and associated large-capacity battery storage systems. A key benefit of the application will be its ability to integrate and centralize control over various devices through a simple web interface, which will provide users with access to real-time data and enable them to perform remote control.

Keywords: programmable controllers, SCADA (Supervisory Control and Data Acquisition), solar power plants, transmission protocols, Python (programming language)

Tímto bych rád poděkoval panu doc. Ing. Bc. Bronislavu Chramcovi, Ph.D. za odborný dohled, cenné rady, přívětivý přístup a konzultace, díky kterým se mi podařilo tuto práci zkompletovat.

Dále bych chtěl na tomto místě poděkovat mé rodině, mým nejbližším a hlavně mé manželce Veronice za podporu, kterou mi poskytovali po celou dobu mého studia.

OBSAH

ÚVOD	7
I TEORETICKÁ ČÁST	8
1 SEZNÁMENÍ S AKTUÁLNÍM STAVEM	9
1.1 ANALÝZA DOHLEDOVÉHO CENTRA FIRMY SUNGROW.....	12
1.1.1 Zobrazení aktuálních dat.....	13
1.1.2 Řízení.....	17
1.2 ANALÝZA DOHLEDOVÉHO CENTRA FIRMY ALPHAESS.....	19
1.2.1 Zobrazení aktuálních dat.....	19
1.2.2 Řízení.....	21
2 ZÁKLADNÍ POJMY	22
2.1 DEFINICE A VYSVĚTLENÍ TERMÍNŮ.....	22
2.1.1 Fotovoltaická elektrárna.....	22
2.1.2 Velkokapacitní baterie.....	22
2.1.3 OTE.....	22
2.2 SOUHRN POUŽITÉ TECHNOLOGIE.....	23
2.2.1 PLC Unipi Patron S107.....	23
2.2.2 Komunikační protokol Modbus.....	24
2.2.3 Programovací jazyk Python 3.....	24
2.2.4 Linux Debian.....	25
II PRAKTICKÁ ČÁST	27
3 NÁVRH STRUKTURY SYSTÉMU PRO SPRÁVU A ŘÍZENÍ FOTOVOLTAICKÝCH A VELKOKAPACITNÍCH BATERIÍ	28
4 POSTUP VÝVOJE APLIKACE PRO MONITORING A ŘÍZENÍ	30
4.1 PŘÍPRAVA VÝVOJOVÉHO PROSTŘEDÍ.....	30
4.2 VYTVOŘENÍ MODULU PRO ZÍSKÁNÍ CENY ELEKTRICKÉ ENERGIE.....	35
4.3 VÝVOJ MONITORINGU A ŘÍZENÍ.....	37
5 POŽADAVKY NA SPOLEHLIVOST A ZABEZPEČENÍ APLIKACE	46
6 PŘÍNOS APLIKACE	48
ZÁVĚR	50
SEZNAM POUŽITÉ LITERATURY	51
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	54
SEZNAM OBRÁZKŮ	55
SEZNAM PŘÍLOH	56

ÚVOD

Téma mé bakalářské práce jsem zvolil s ambicí vyvinout unikátní aplikaci, která má za úkol usnadnit sběr dat a následné řízení fotovoltaických elektráren. To by mělo v budoucnosti přinést vlastníkům fotovoltaických elektráren výhodu v podobě centralizovaného dohledového centra, což povede ke zjednodušení správy a potenciální finanční úspoře. Vlastník bude mít možnost přidávat libovolné značky fotovoltaických střídačů do stávající instalace podle svých preferencí nebo rozpočtu. Díky centralizované správě bude možné fotovoltaické elektrárny buď manuálně, nebo automaticky vypínat v případě, že výkupní cena elektřiny klesne do záporných hodnot, což vše bude řízeno z jediného místa.

V současnosti je optimalizace výroby a distribuce solární energie značně diskutovanou problematikou, neboť v době nejvyšší produkce solární energie může dojít k nadbytku elektrické energie, což může vést k záporným cenám výkupu elektrické energie. Tato situace představuje finanční zátěž pro provozovatele fotovoltaických elektráren, neboť jsou penalizováni za dodávání elektrické energie do distribuční sítě. Mé řešení spočívá ve vývoji aplikace, která bude monitorovat tržní ceny výkupu elektrické energie a při hrozící záporné ceně automaticky vypne fotovoltaickou elektrárnu nebo zamezí dodávku elektrické energie velkokapacitní baterii. Aplikace bude navržena tak, aby dokázala integrovat různé typy fotovoltaických střídačů a velkokapacitních baterií, což umožní efektivní sběr dat a správu zařízení. Centralizovaná správa a řízení z jediného dohledového centra tak nabídnou komplexní řešení pro optimalizaci provozu fotovoltaických elektráren a velkokapacitních baterií.

Výsledek této práce představuje robustní základy pro vývoj komplexního dohledového centra, které bude schopné automaticky řídit i instalace fotovoltaických elektráren spojené s velkokapacitními bateriemi, což přináší další možnosti řízení a automatizace. Aplikace je však koncipována tak, aby poskytovala pevný základ pro budoucí rozšiřování o další funkcionality.

I. TEORETICKÁ ČÁST

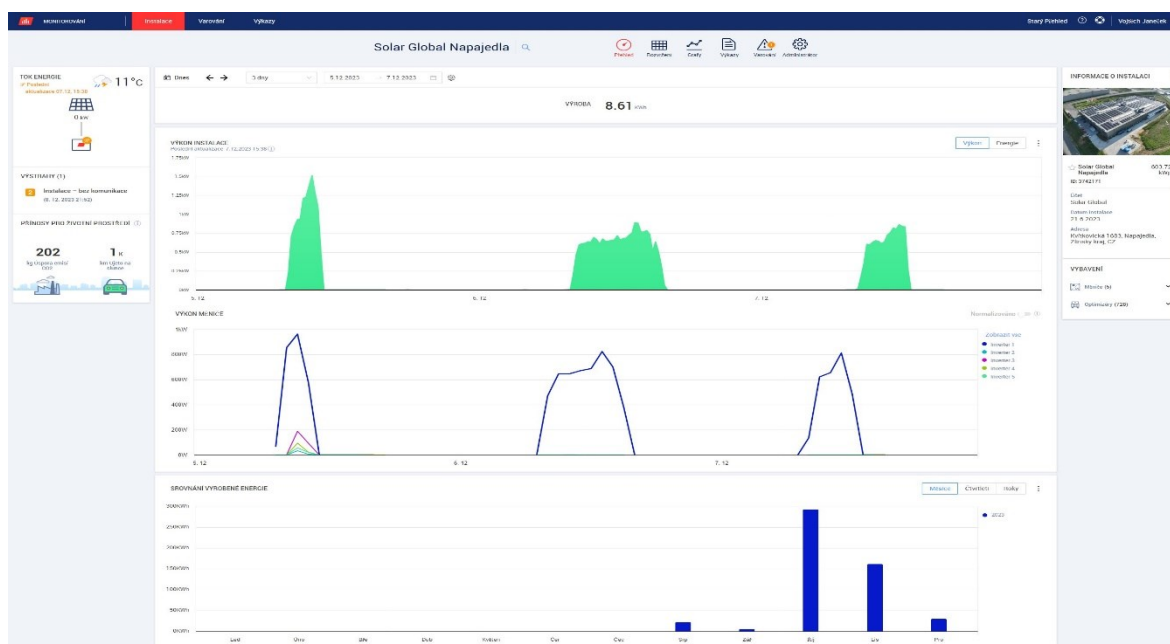
1 SEZNÁMENÍ S AKTUÁLNÍM STAVEM

Současný stav dohledových center fotovoltaických elektráren a velkokapacitních baterií můžeme charakterizovat jako pokročilý a sofistikovaný systém, který umožňuje monitorování a řízení výkonu těchto zařízení v reálném čase. Klíčové prvky tohoto systému zahrnují monitorování výkonu, sběr a analýzu dat, diagnostiku a vzdálené řízení.

Mezi významné nedostatky patří omezení možnosti propojení různých výrobců fotovoltaických střídačů, což komplikuje celkový dohled nad systémem a snižuje jeho interoperabilitu. Dalším nedostatkem je v některých případech úplné omezení možnosti vzdáleného ovládání, což brání možnosti dynamického omezování výkonu fotovoltaické elektrárny v závislosti na aktuální ceně elektrické energie, pokud není použito dalších speciálních zařízení. Taková omezení mohou omezovat flexibilitu a efektivitu provozu fotovoltaických elektráren.

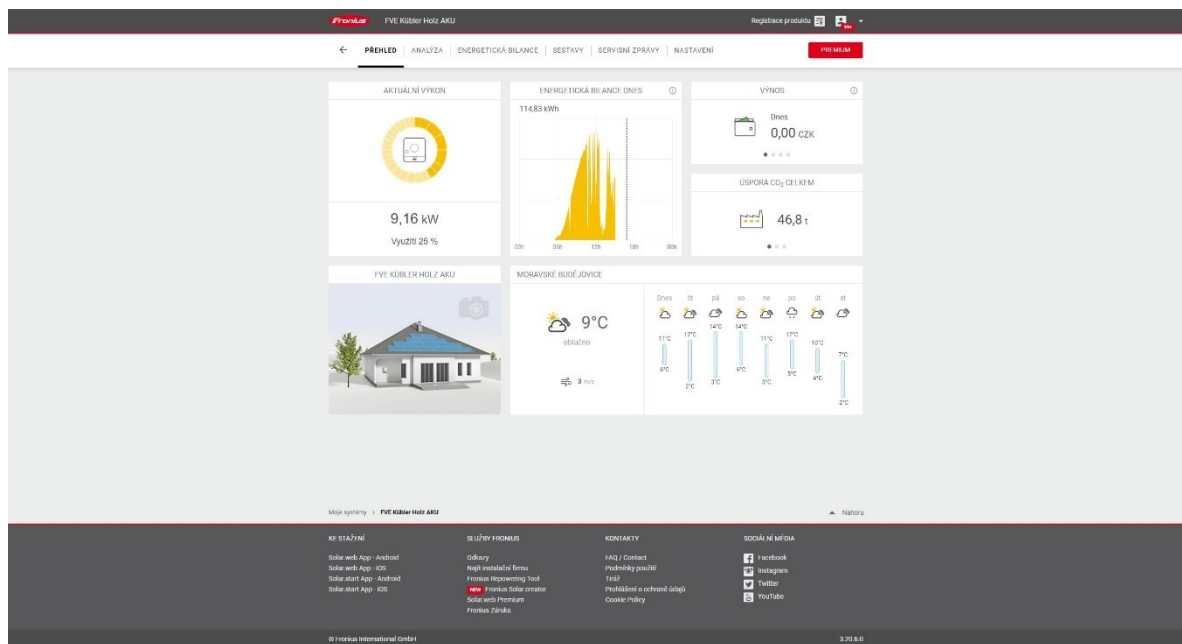
Pro získání komplexního přehledu o struktuře dohledových center v oblasti fotovoltaických instalací jsem se zaměřil na reprezentativní vzorek od předních výrobců fotovoltaických střídačů.

Dohledové centrum společnosti SolarEdge je charakterizováno základními informacemi o výrobním výkonu, bezpečnostními upozorněními, předpovědi počasí a historickými daty týkajícími se provozu jednotlivých střídačů. Je důležité poznamenat, že v tomto prostředí chybí funkcionality vzdáleného ovládání fotovoltaických střídačů, což může omezovat možnosti monitoringu a řízení energetických systémů.



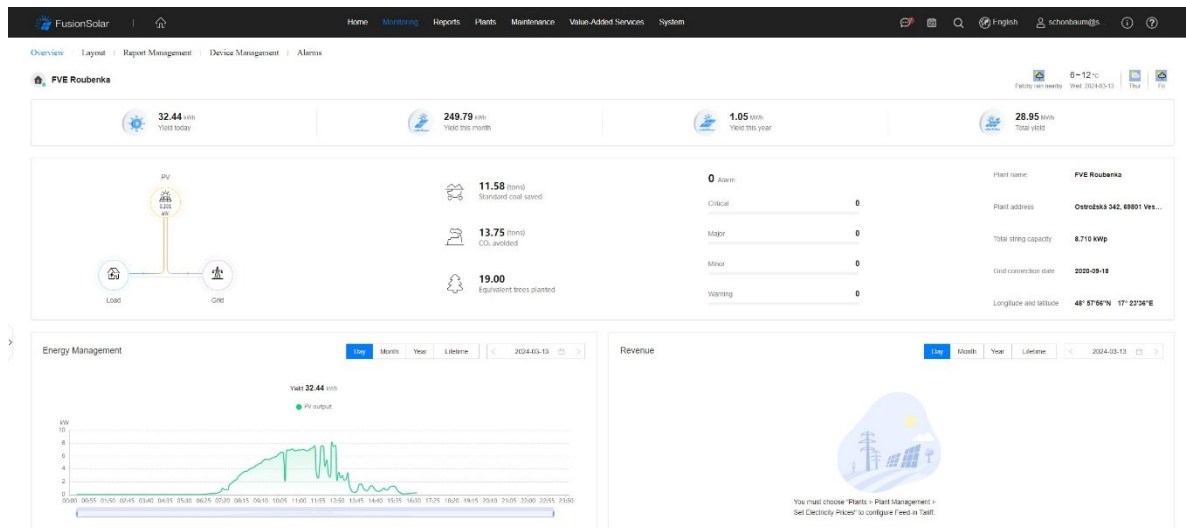
Obrázek 1. Dohledové centrum firmy SolarEdge [1]

Jako další exemplář lze uvést dohledové centrum od výrobce fotovoltaických střídačů Fronius. Toto centrum poskytuje srovnatelnou funkcionalitu s dohledovým centrem společnosti SolarEdge, nabízející zobrazení aktuálního výkonu, historických dat, servisních zpráv a předpovědi počasí. Avšak, stejně jako v předchozím případě, zde není dostupná možnost vzdáleného řízení fotovoltaických střídačů.

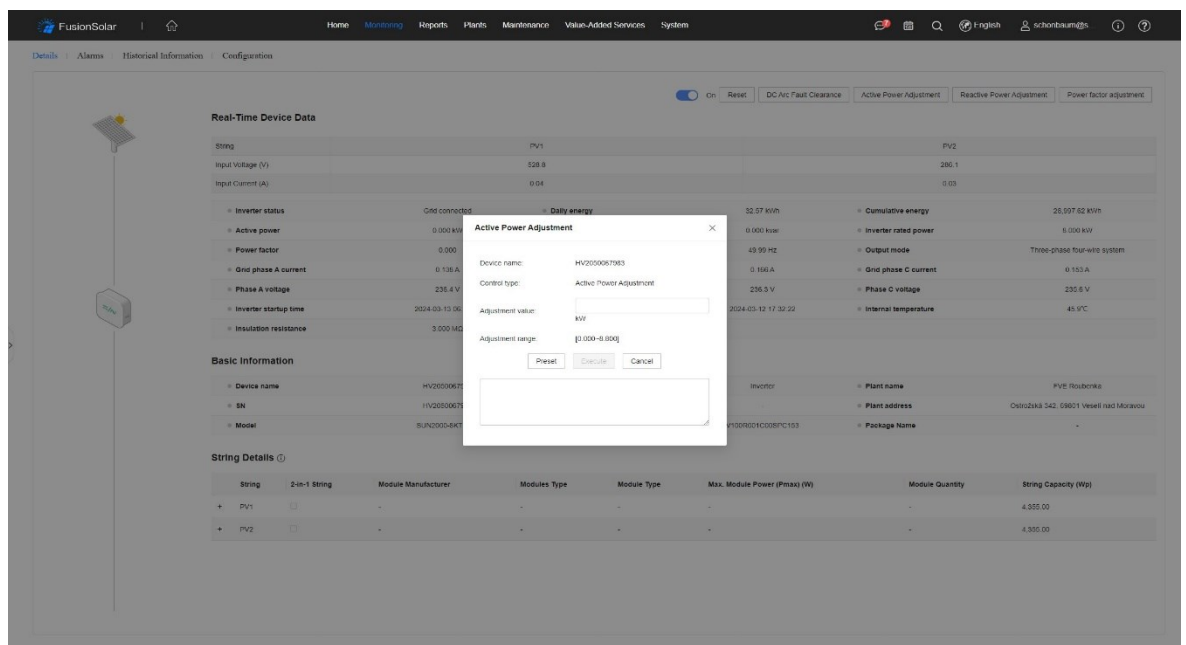


Obrázek 2. Dohledové centrum firmy Fronius [2]

Dohledová centra od společností Huawei a Sungrow přinášejí rozšířené funkce oproti předchozím příkladům. Dohledové centrum od firmy Huawei umožňuje nejen zobrazování základních informací jako již známe z dohledových center SolarEdge a Fronius, ale také umožňuje vzdálené ovládání zařízení. Tato funkce je ovšem umístěna méně přehledně pod záložkou "Device Management". Po přechodu na stránku „Device Management“ uživatel musí najít a rozkliknout příslušný střídač, a teprve poté pomocí tlačítka "Active Power Adjustment" v pravém horním rohu nastavit hodnotu maximálního výkonu, jak je znázorněno na Obrázku 4. Dohledové centrum od společnosti Sungrow si detailněji popíšeme v další části práce.



Obrázek 3. Dohledové centrum firmy Huawei [3]

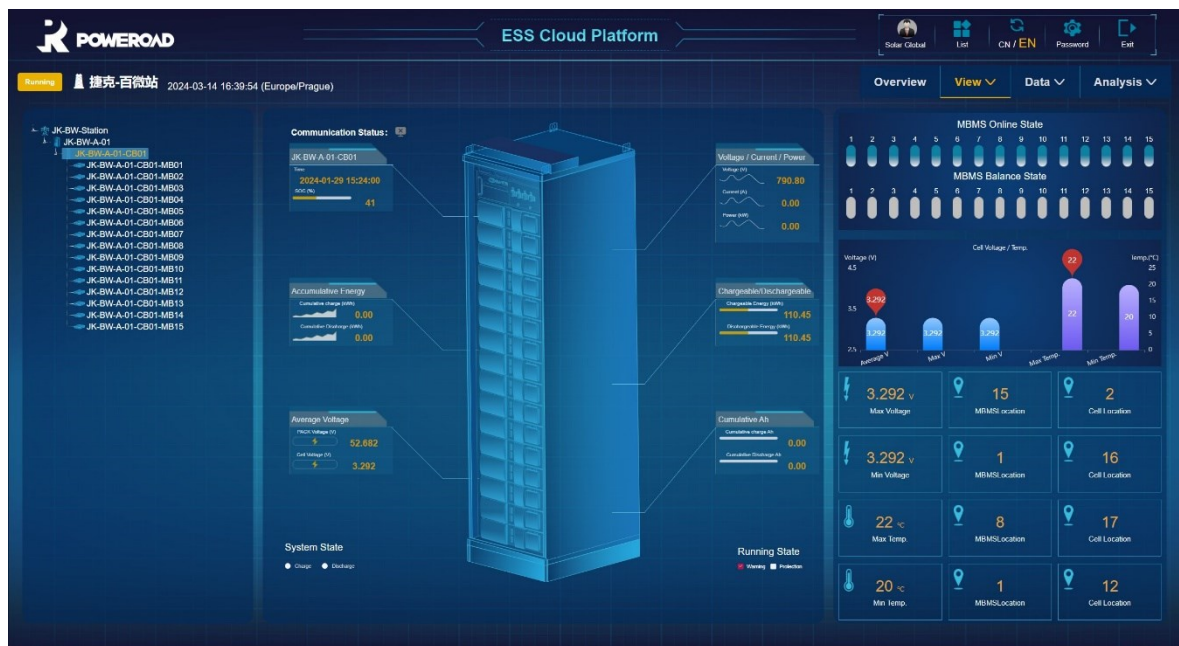


Obrázek 4. Dohledové centrum firmy Huawei – nastavení maximálního výkonu střídače [3]

V současném kontextu fotovoltaických elektráren se stává stále důležitějším prvkem velkokapacitní bateriové uložení. Tyto uložení hrají klíčovou roli v optimalizaci využití vyrobené

energie a zajištění stability energetických sítí. Pro monitorování a řízení těchto systémů se využívají dohledová centra, která umožňují sledovat a optimalizovat provoz uložišť.

V rámci této práce byla provedena analýza dohledových center od předních výrobců jako jsou společnost Poweroad a společnost AlphaESS. Z provedeného výběru vychází, že dohledové centrum firmy Poweroad zatím nenabízí plnou funkcionalitu vzdáleného ovládání bateriového uložiště. Uživatelé jsou omezeni pouze na sledování aktuálních a historických hodnot bateriového uložiště prostřednictvím tohoto rozhraní. Přesto však tato platforma poskytuje komplexní informace, jako je například možnost monitorovat napětí na úrovni jednotlivých bateriových článků. Dále umožňuje zobrazovat informace o stavu nabití, aktuálním výkonu, teplotě, alarmech, napětí a prouděch bateriového uložiště, což přispívá k efektivnímu řízení a údržbě těchto systémů.



Obrázek 5. Dohledové centrum firmy Poweroad [4]

V porovnání s dohledovým centrem firmy Poweroad, dohledové centrum firmy AlphaESS poskytuje širší funkcionalitu v podobě možnosti vzdáleného ovládání, což představuje významný pokrok v oblasti monitorování a řízení bateriových uložišť. Toto dohledové centrum si proto budeme detailně analyzovat v další části práce.

1.1 Analýza dohledového centra firmy Sungrow

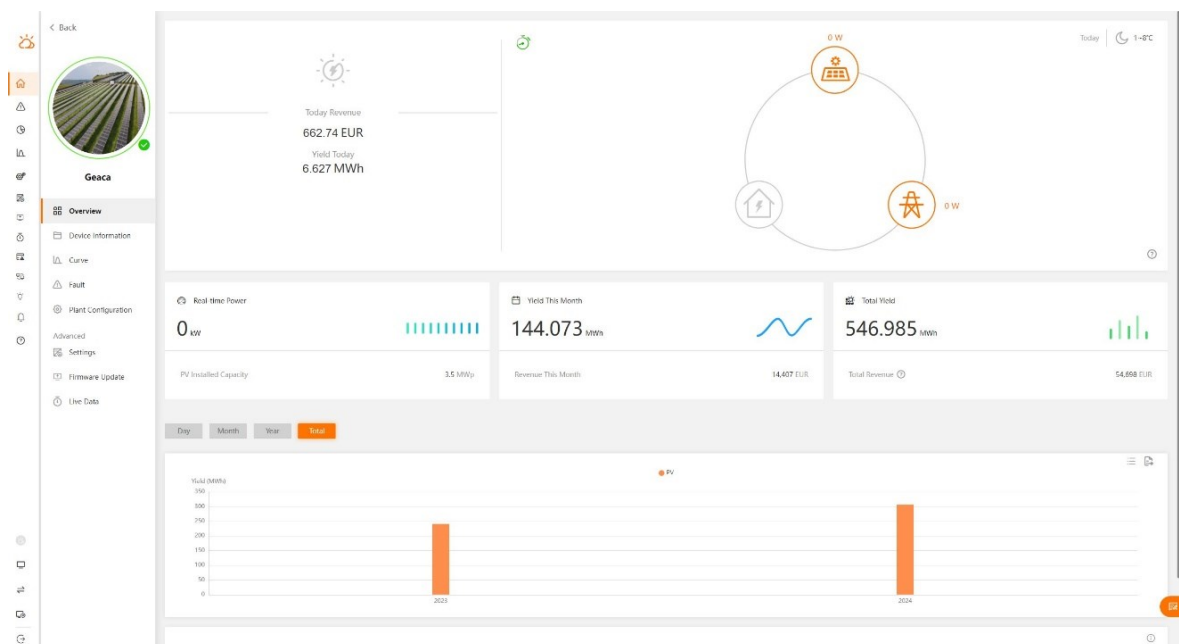
Jako příklad pro detailnější popis dohledového centra jsem vybral systém od firmy Sungrow, který poskytuje uživatelům přístup k aktuálním datům a umožňuje vzdálené řízení

fotovoltaických elektráren. Tento systém je navržen tak, aby poskytoval komplexní monitorování a správu výkonu solárních elektráren v reálném čase.

Dohledové centrum slouží k monitorování aktuálních dat souvisejících s provozem solárních panelů a střídačů. Tato data zahrnují informace o aktuálním výkonu, celkově vyrobené energii, aktuálním stavu počasí a historii těchto naměřených hodnot. Tyto údaje jsou klíčové pro operátory, kteří mají za úkol efektivně řídit provoz solárních elektráren.

1.1.1 Zobrazení aktuálních dat

V případě dohledového centra od společnosti Sungrow, zobrazeného na Obrázku 6, jsou klíčové informace umístěny v levém horním rohu, kde je zobrazena hodnota celkové denní výroby elektrické energie. V pravém horním rohu se nachází v reálném čase se měnící graf proudění vyrobené energie. Pod touto hodnotou jsou tři bloky obsahující aktuální údaje o výkonu, vyrobené energii v průběhu aktuálního měsíce a celkově vyrobené energii od počátku instalace. V dolní části obrazovky pak můžeme nalézt zobrazení historických dat, což poskytuje užitečný kontext pro analýzu provozu solárních elektráren v čase.

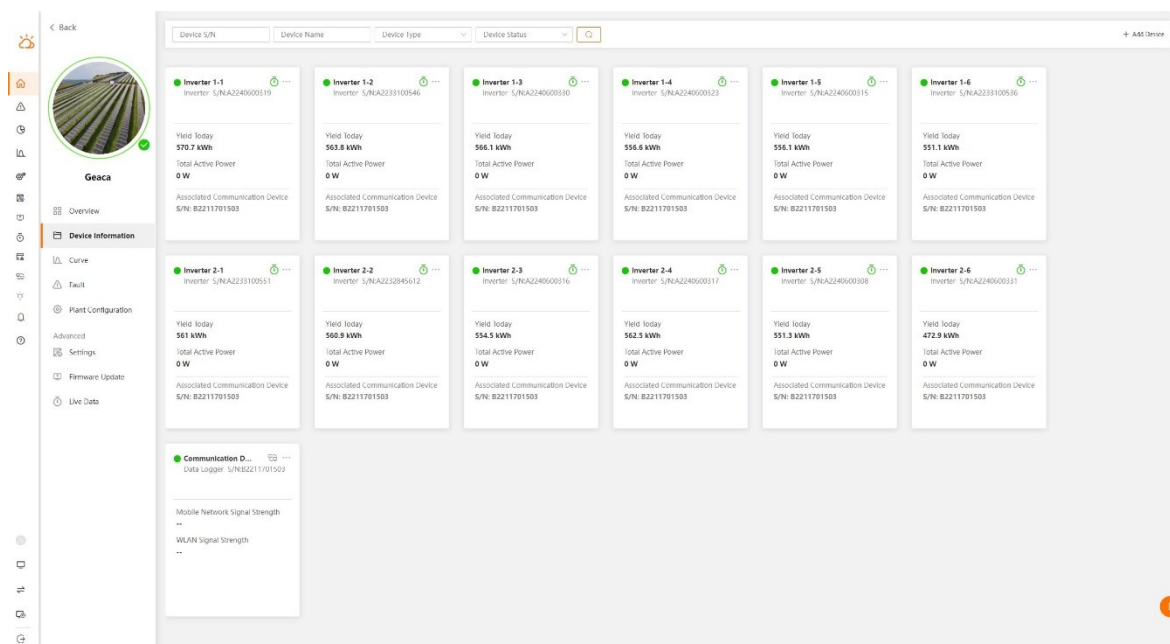


Obrázek 6. Dohledové centrum firmy Sungrow [5]

V sekci "Device Information" v bočním menu, jak je znázorněno na Obrázku 7, existuje možnost získat detailní informace o jednotlivých střídačích. Tato data jsou zvláště cenná při diagnostice závad na konkrétních střídačích. Po rozkliknutí střídače jsou zobrazeny důležité informace, jako je napětí a proudy na jednotlivých fázích, hodnota izolačního odporu nebo proudy na stringzích. Nicméně je třeba zdůraznit, že většina zobrazených hodnot je pro

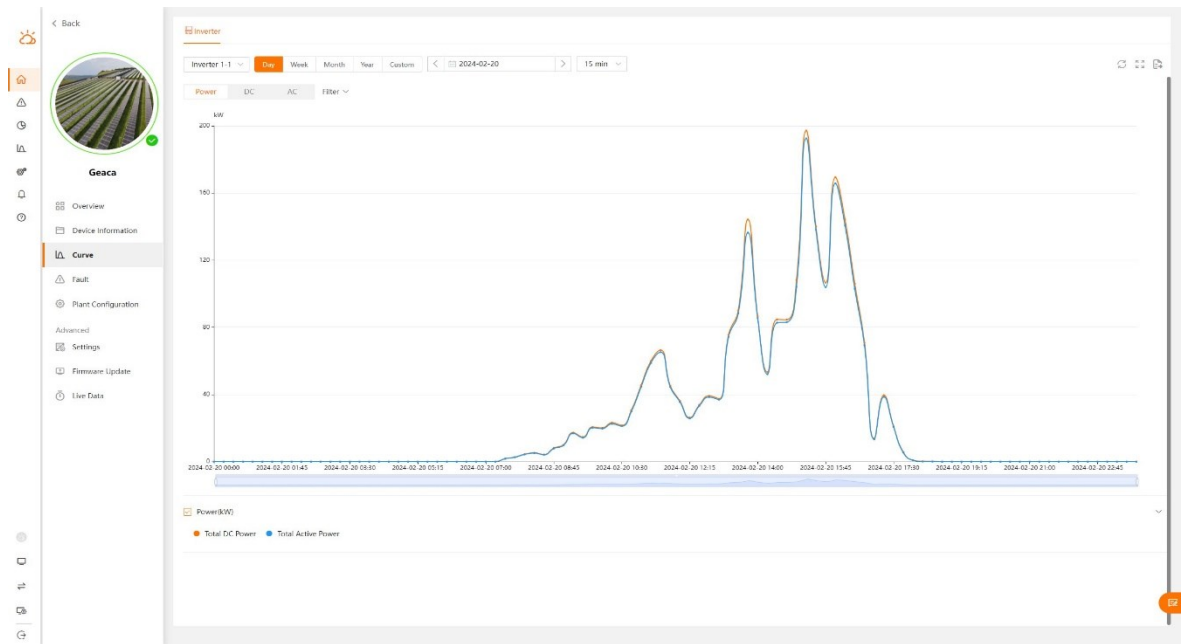
běžného uživatele druhořadá a při mnoha příležitostech může být takové množství informací spíše matoucí.

Pro účely efektivního a uživatelsky přívětivého provozu dohledového systému je důležité zvážit, které informace jsou pro uživatele relevantní a důležité. Přílišná komplexnost a nadměrné množství dat mohou vést k nepochopení a ztrátě přehledu. Proto je důležité přizpůsobit prezentované informace potřebám a schopnostem uživatelů, aby byla zajištěna efektivní a přehledná správa solárních elektráren.



Obrázek 7. Dohledové centrum firmy Sungrow – sekce „Device Information“ [5]

V rámci sekce nazvané "Curve" znázorněné na Obrázku 8. Máme přístup k historickým datům ze střídačů, která jsou zobrazena v grafické podobě. Uživatel má možnost vybírat konkrétní data z určitého časového období.



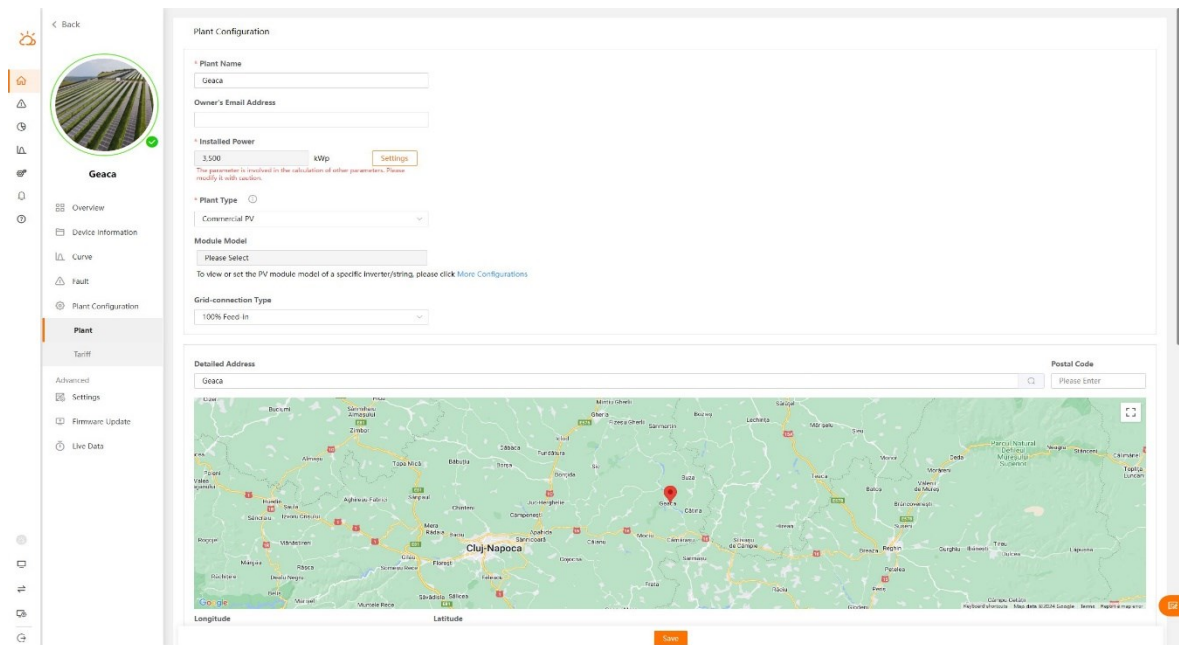
Obrázek 8. Dohledové centrum firmy Sungrow - sekce "Curve" [5]

V sekci nazvané "Fault" znázorněné na Obrázku 9. Jsou zobrazena individuální chybová hlášení pocházející ze střídačů. Uživatel má opět možnost definovat vlastní časové období pro zpětnou diagnostiku chyb.

Alarm type	Fault Code	Alarm Name	Device Name	Device Model	Occurrence Time	Action
No Data						

Obrázek 9. Dohledové centrum firmy Sungrow - sekce "Fault" a chybová hlášení střídačů [5]

V sekci nazvané "Plant Configuration" znázorněné na Obrázku 10. Je možné provádět konfiguraci instalace, včetně nastavení názvu instalace, emailové adresy vlastníka, kam jsou zasílány informační emaily v případě jakéhokoli změny na dohledovém centru, typu instalace, typu připojení do sítě a přesné polohy instalace.



Obrázek 10. Dohledové centrum firmy Sungrow - sekce "Plant Configuration" [5]

V dalších sekcích „Settings“, „Firmware Update“ a „Live Data“ zobrazených na Obrázku 11. Je uživateli poskytnuta možnost provádět konfiguraci ochran pro připojení střídačů do distribuční sítě, aktualizovat firmware a zobrazit živá produkční data. Tato funkcionality zobrazování živých dat je velmi podobná již zmíněnému zobrazení dat v sekci „Device Information“.

Inverter	Device Model	Country/Regi...	Device S/N	Template Library	Initial Grid Connection	Common Parameter Settings	Task List		
Plant Name	Device Name	Initial Grid	Device S/N	Device Model	Country/Region	Grid Type	Version No.	Device Location	Action
Geaca	Inverter 1-1	Already Set	A2240900319	SG250HX	EN5049-2	50 Hz	CS1-2.0.1.35-AS1-1.1.25.0-AA10-1.0.4.0	Grid-connected point 1.14unit	🔍 📄 🔄
Geaca	Inverter 2-6	Already Set	A2240900331	SG250HX	EN5049-2	50 Hz	CS1-2.0.1.35-AS1-1.1.25.0-AA10-1.0.4.0	Grid-connected point 1.14unit	🔍 📄 🔄
Geaca	Inverter 2-5	Already Set	A2240900308	SG250HX	EN5049-2	50 Hz	CS1-2.0.1.35-AS1-1.1.25.0-AA10-1.0.4.0	Grid-connected point 1.14unit	🔍 📄 🔄
Geaca	Inverter 2-1	Already Set	A2233100551	SG250HX	EN5049-2	50 Hz	CS1-2.0.1.35-AS1-1.1.25.0-AA10-1.0.4.0	Grid-connected point 1.14unit	🔍 📄 🔄
Geaca	Inverter 1-2	Already Set	A2233100546	SG250HX	EN5049-2	50 Hz	CS1-2.0.1.35-AS1-1.1.25.0-AA10-1.0.4.0	Grid-connected point 1.14unit	🔍 📄 🔄
Geaca	Inverter 1-5	Already Set	A2240900315	SG250HX	EN5049-2	50 Hz	CS1-2.0.1.35-AS1-1.1.25.0-AA10-1.0.4.0	Grid-connected point 1.14unit	🔍 📄 🔄
Geaca	Inverter 2-3	Already Set	A2240900316	SG250HX	EN5049-2	50 Hz	CS1-2.0.1.35-AS1-1.1.25.0-AA10-1.0.4.0	Grid-connected point 1.14unit	🔍 📄 🔄
Geaca	Inverter 1-6	Already Set	A2233100536	SG250HX	EN5049-2	50 Hz	CS1-2.0.1.35-AS1-1.1.25.0-AA10-1.0.4.0	Grid-connected point 1.14unit	🔍 📄 🔄
Geaca	Inverter 2-4	Already Set	A2240900317	SG250HX	EN5049-2	50 Hz	CS1-2.0.1.35-AS1-1.1.25.0-AA10-1.0.4.0	Grid-connected point 1.14unit	🔍 📄 🔄
Geaca	Inverter 1-4	Already Set	A2240900323	SG250HX	EN5049-2	50 Hz	CS1-2.0.1.35-AS1-1.1.25.0-AA10-1.0.4.0	Grid-connected point 1.14unit	🔍 📄 🔄

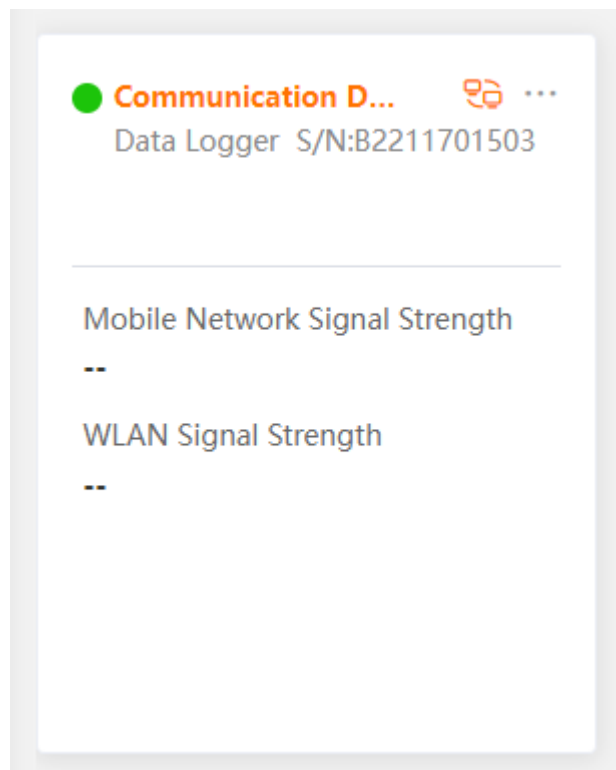
Obrázek 11. Dohledové centrum firmy Sungrow - sekce „Settings“ [5]

1.1.2 Řízení

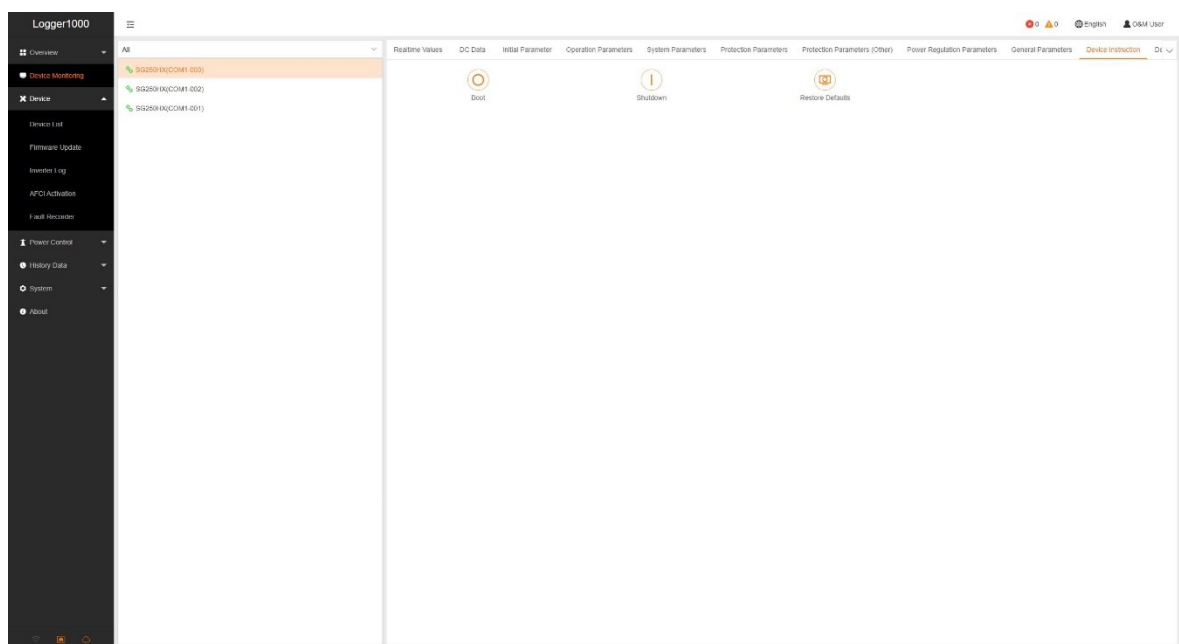
Řízení fotovoltaické elektrárny společnosti Sungrow je realizováno prostřednictvím vzdáleného přístupu ke komunikačním zařízením, jak je znázorněno na Obrázku 12. pomocí ikony dvou vzájemně komunikujících počítačů v pravém horním rohu. Tento přístup umožňuje operátorovi připojení k individuálním fotovoltaickým elektrárnám vzdáleně prostřednictvím komunikačního zařízení a následné manuální ovládání fotovoltaické elektrárny. Manuální ovládání je dostupné v levém sloupci pod položkou „Device Monitoring“ a následně v horní liště pod záložkou „Device Instruction“ na této záložce je již zobrazena možnost manuálního vypnutí nebo v opačném případě zapnutí fotovoltaických střídačů jak je znázorněno na Obrázku 13.

Je třeba zdůraznit, že takové připojování je často zdlouhavé a pro uživatele náročné. Při provozu více elektráren se navíc uživatel musí manuálně připojovat ke každé fotovoltaické elektrárně zvlášť. Avšak i přes tyto obtíže tento postup umožňuje operátorovi provádět řízení a monitorování fotovoltaických elektráren z jakéhokoli místa s připojením k internetu, což přináší větší flexibilitu a efektivitu v provozu fotovoltaických systémů. Takto operátor může

rychle reagovat na případné problémy nebo optimalizovat provoz podle aktuálních potřeb a podmínek.



Obrázek 12. Dohledové centrum firmy Sungrow - Vzdálený přístup pro řízení [5]



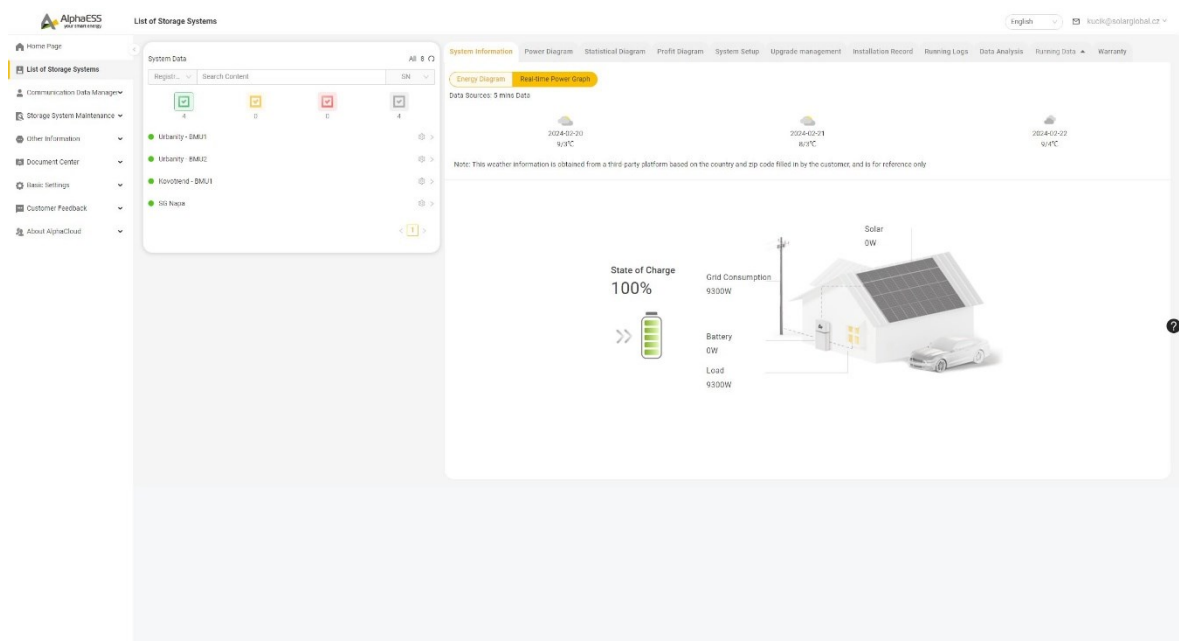
Obrázek 13. Dohledové centrum firmy Sungrow - manuální ovládání fotovoltaických střídačů [5]

1.2 Analýza dohledového centra firmy AlphaESS

Pro detailnější ilustraci současného řešení dohledového centra pro bateriové úložiště jsem zvolil systém od firmy AlphaESS. Tato firma je čínským dodavatelem bateriových úložišť, který má více než desetiletou zkušenost na trhu. Jejich dohledové centrum poskytuje uživatelům komplexní monitorování a správu výkonu a stavu bateriových úložišť, což umožňuje efektivní provoz a optimalizaci využití elektrické energie z těchto systémů.

1.2.1 Zobrazení aktuálních dat

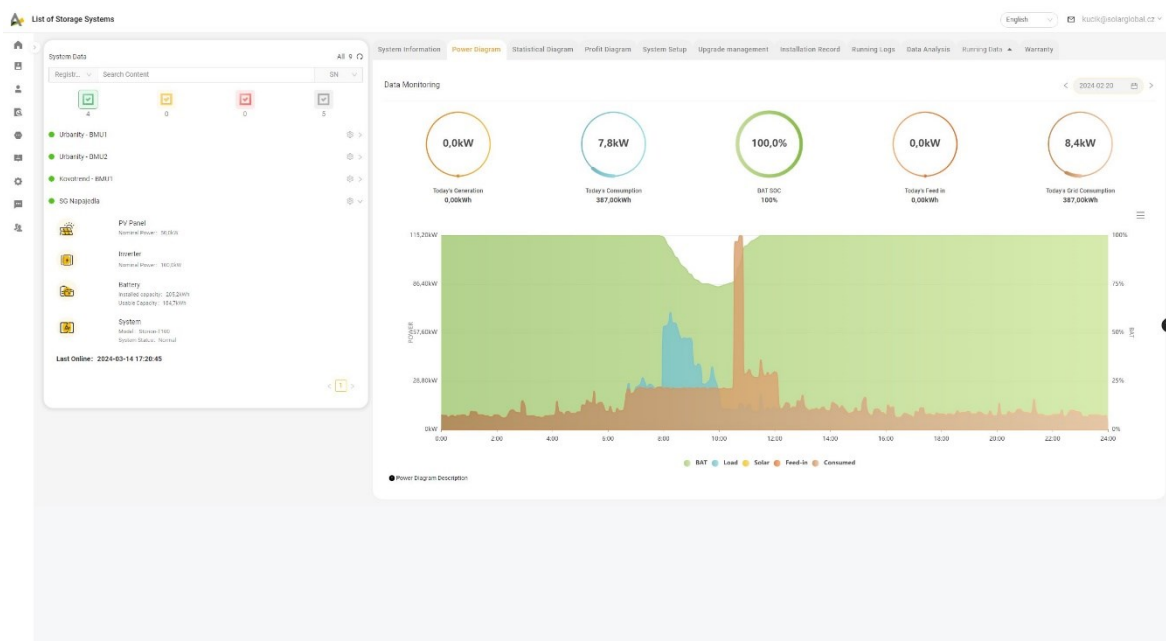
Na úvodní stránce dohledového centra v sekci „System Information“ jsou zobrazeny klíčové parametry související s provozem bateriového úložiště. Uživatel zde může sledovat aktuální výkon střídače, informace o dobíjení nebo vybíjení baterie, aktuální stav nabití úložiště a směr toku energie. Směr toku energie je zobrazen ve vztahu k distribuční síti, tj. uživatel má možnost sledovat, zda se energie právě odebírá ze sítě do úložiště nebo naopak, zda se energie z úložiště vrací do distribuční sítě. Tyto informace umožňují uživatelům monitorovat a optimalizovat provoz svého bateriového úložiště v souladu s aktuálními potřebami a podmínkami.



Obrázek 14. Dohledové centrum firmy AlphaESS [6]

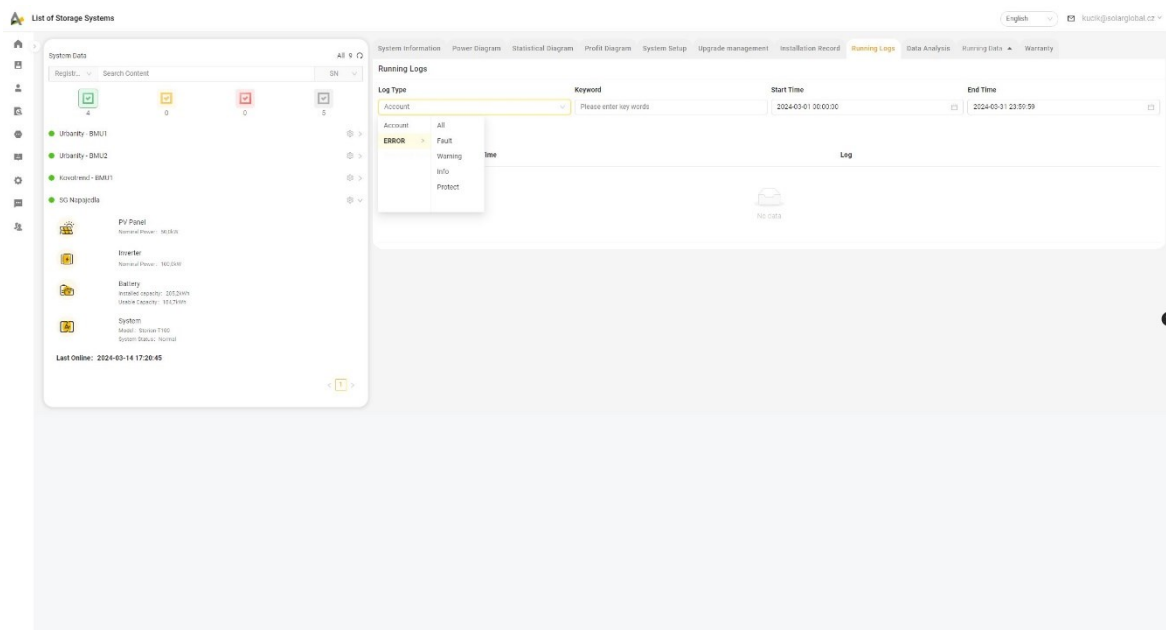
V rámci sekce nazvané „Power Diagram“ má uživatel možnost sledovat historická data, jako je stav nabití baterie, pokrytí odběru areálu, ve kterém je baterie instalována, dodávku do

sítě a v případě, že je baterie kombinována s fotovoltaickou elektrárnou, také výkon této elektrárny.



Obrázek 15. Dohledové centrum firmy AlphaESS - sekce "Power Diagram" [6]

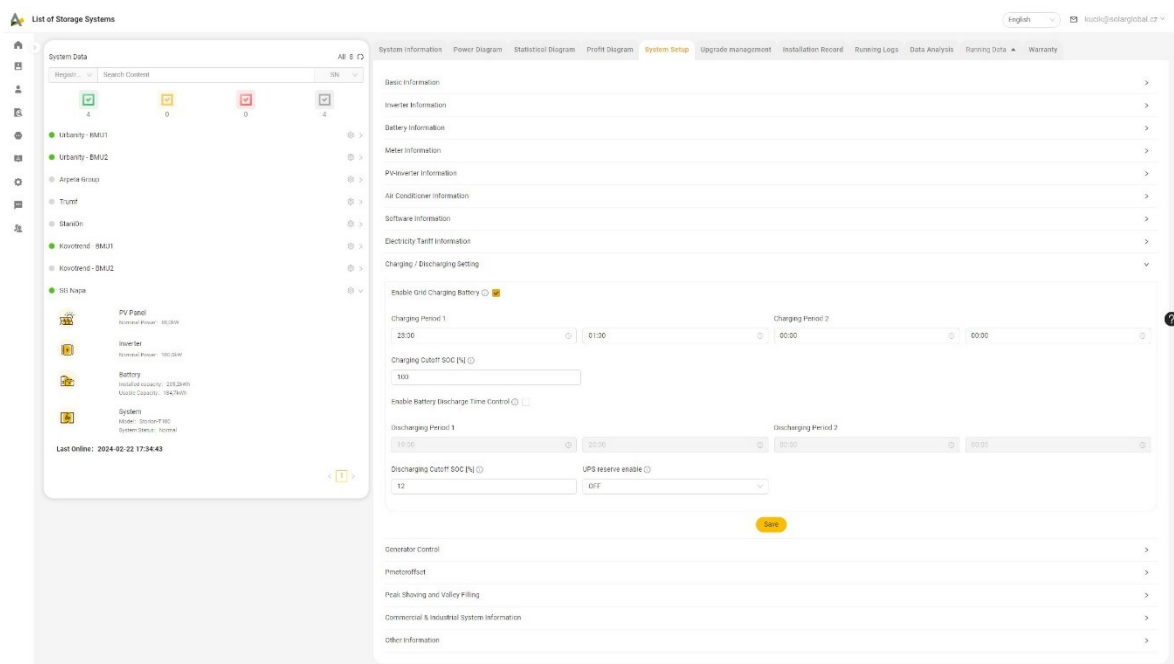
Další relevantní sekce, která zasluží pozornost, je sekce nazvaná „Running Logs“, kde uživatel může nalézt historické záznamy chyb a poruch. Uživatel má možnost tyto záznamy filtrovat podle jejich důležitosti, což usnadňuje identifikaci a řešení problémů s bateriovým uložištěm.



Obrázek 16. Dohledové centrum firmy AlphaESS - sekce "Running Logs" [6]

1.2.2 Řízení

Řízení bateriového úložiště je přístupné prostřednictvím záložky "System Setup", jak je patrné na Obrázku 17, kde kromě nastavení řízení jsou uvedeny další informace o úložišti jako je výrobní číslo, modelové označení střídače nebo aktuální verze firmwaru. Nicméně pro koncového uživatele mohou být tyto informace pouze marginálního významu. Při přechodu do podzáložky "Charging/Discharging Battery" jsou zobrazeny možnosti nastavení dvou časových intervalů pro nabíjení a vybíjení baterie. Tato konfigurace však může pro uživatele představovat výzvu, neboť vyžaduje ruční úpravy předem. Například při změně výkupní ceny elektrické energie je nezbytné manuálně upravit tyto hodnoty, což může být pro uživatele složité a časově náročné. Mimo toto manuální ovládání baterie poskytuje mód, který umožňuje regulaci přetoků elektrické energie do distribuční sítě nebo mód, který může vykrývat výkonové špičky odběru areálu, kde je instalována.



Obrázek 17. Dohledové centrum firmy AlphaESS – sekce „System Setup“ [6]

2 ZÁKLADNÍ POJMY

Při vytváření aplikace, která se zaměřuje na energetiku, je klíčové seznámit se s širokým spektrem termínů, jež v této oblasti převažují, zejména pokud jde o obnovitelné zdroje energie. Tato oblast je plná odborných výrazů a konceptů, které jsou nezbytné pro porozumění a efektivní implementaci aplikace. Důležité je také porozumět technickým termínům týkajícím se distribuce, skladování a využití těchto zdrojů, abychom mohli aplikaci úspěšně navrhnout a optimalizovat pro potřeby uživatelů v oblasti obnovitelné energetiky.

2.1 Definice a vysvětlení termínů

2.1.1 Fotovoltaická elektrárna

Fotovoltaická elektrárna představuje efektivní způsob získávání elektrické energie přímo ze slunečního záření pomocí fotovoltaického jevu na fotodiodách. Tyto fotodiody jsou spojovány do velkých formátů známých jako fotovoltaické panely. Jejich úkolem je zachytit sluneční záření a přeměnit ho na elektrickou energii. Aby byla tato energie použitelná v distribuční síti, je nezbytné převést jednosměrné napětí produkované fotovoltaickými panely na střídavé napětí. K tomu slouží zařízení nazývaná fotovoltaické střídače, která transformují výstupní jednosměrné napětí na střídavé napětí vhodné pro použití v distribuční síti. Takto fungující fotovoltaická elektrárna je ekologicky šetrným zdrojem energie a přispívá k udržitelnému rozvoji. [7]

2.1.2 Velkokapacitní baterie

Velkokapacitní baterie představují seskupení menších baterií do jednoho velkého celku, který slouží k uchování elektrické energie. Tento typ baterie se skládá z střídače, který transformuje stejnosměrný proud z baterií na střídavý proud, vhodný pro distribuční síť. Bateriové uložení, známé též pod zkratkou BESS (Battery Energy Storage System), patří mezi nejrychleji reagující zdroje energie v elektrických sítích. To znamená, že baterie může během několika vteřin přejít z režimu připravenosti do maximálního výkonu střídače. Tato funkce je často využívána k vyrovnávání frekvencí v distribuční síti, ať už je potřeba zvýšit či snížit frekvenci. [8]

2.1.3 OTE

OTE jako akciová společnost ve vlastnictví České republiky a také operátorem trhu, který má licenci č.150505700 udělenou Energetickým regulačním úřadem. Její hlavní činnosti

zahrnují organizaci krátkodobého trhu s plynem a elektřinou, vyhodnocování a zúčtování odchylek v celé České republice, poskytování informací o platebních povinnostech účastníků trhu a správu registru obchodování s povolenkami na emise skleníkových plynů. Navíc poskytuje služby jako zpracování a zveřejňování tržních zpráv, správu podkladů pro pravidla trhu s energií, zajišťování skutečných hodnot dodávek a odběrů energie a administraci systémů pro vyplácení podpory a správu záruk původu. Celkově se firma specializuje na řízení a správu tržních procesů a infrastruktury v energetickém sektoru. [9]

2.2 Souhrn použité technologie

2.2.1 PLC Unipi Patron S107

Jako hlavní prvek celé aplikace můžeme označit programovatelný logický kontrolér (PLC) Unipi Patron S107. Toto PLC od brněnského výrobce je navrženo pro automatizaci, ovládání, regulaci i monitoring a odpovídá požadavkům pro instalaci v průmyslovém odvětví. Ze základních parametrů můžeme vyzvednout čtyřjádrový ARMový procesor Cortex-A53, 1GB RAM, čtyři digitální vstupy/výstupy, jeden analogový vstup/výstup, tři sérová rozhraní a ethernetový port. Pro naše účely je také nutné zmínit, že PLC disponuje softwarovou otevřeností jelikož je založeno na operačním systému Linux. [10]

Nedílnou součástí procesu programování PLC Unipi Patron S107 je využití vývojového prostředí nazvaného Mervis IDE. Toto prostředí je navrženo speciálně pro tvorbu, ladění a vzdálenou správu programů, které jsou určeny pro kontroléry Unipi. Jeho flexibilita spočívá v možnosti využití dvou programovacích metod, které odpovídají standardu IEC61131-3. [11]

První z těchto metod je založena na programování pomocí funkčních bloků (FUPLA), které jsou někdy označovány také jako FBD bloky. Tento přístup umožňuje rychlou a intuitivní tvorbu programů, jelikož bloky již obsahují předdefinované vstupy a výstupy. Programátor tak pouze propojuje tyto bloky a nastavuje potřebné parametry. Mezi předdefinované FBD bloky patří například matematické operace, řídicí smyčky nebo časové funkce. [11]

Druhou metodou programování je využití strukturovaného textu (ST), což je vhodné zejména pro zkušenější programátory. Jazyk ST umožňuje psát složitější funkce na menší počet řádků, což přispívá k efektivitě vytvářených programů. Pokročilí uživatelé také mají možnost vytvářet vlastní FBD bloky pomocí programování v jazyce ST. [11]

Kromě tvorby samotných programů umožňuje Mervis IDE také vytváření grafického uživatelského rozhraní pro lokální webové stránky nebo pro vzdálený dispečink Mervis SCADA. Tento editor disponuje bohatou paletou prvků, jako jsou přepínače, indikátory, textová pole a další, což umožňuje uživatelům vytvářet přehledné a intuitivní ovládací panely pro své aplikace.

Mervis SCADA představuje cloudovou službu, která umožňuje vzdálený dohled, řízení a analýzu dat z různých zařízení, bez ohledu na jejich umístění. Pro zajištění bezpečného přístupu k PLC jednotkám z libovolného místa a následného ukládání dat na Mervis SCADA slouží Mervis Proxy, což je zabezpečené internetové spojení. Dále je k dispozici cloudová databáze Mervis DB, která umožňuje ukládání historických dat v bezpečném prostředí datového centra společnosti Unipi. [12]

2.2.2 Komunikační protokol Modbus

Protokol Modbus představuje komunikační strukturu, kterou vyvinula společnost Modicon. Je využíván k navázání komunikace mezi klientem a serverem v průmyslovém prostředí. Tento protokol se řadí mezi nejrozšířenější komunikační protokoly v průmyslovém sektoru. V roce 1999 byla vyvinuta otevřená specifikace Modbus TCP/IP, což umožnilo kombinaci tohoto protokolu s univerzálním síťovým standardem TCP/IP. Díky této implementaci je možné navázat komunikaci s jakýmkoli zařízením podporujícím TCP/IP sockety v rámci škálovatelných a všestranných fyzických sítí. [13]

2.2.3 Programovací jazyk Python 3

Programovací jazyk Python 3, který vyvinul Guido van Rossum v roce 1999, se vyznačuje snadnou naučitelností a vysokou úrovní výkonu. Nabízí efektivní datové struktury a jednoduchý, avšak účinný přístup k objektově orientovanému programování. Jeho elegantní syntaxe a dynamický charakter spolu s interpretovanou povahou z něj činí ideální volbu pro skriptování a rychlý vývoj aplikací v široké škále oblastí a na různých platformách. Díky multiplatformní povaze je Python vhodný pro široké spektrum aplikací.

Python je snadno rozšiřitelný o různé knihovny třetích stran, což z něj činí flexibilní nástroj pro řešení různorodých problémů v různých oblastech vývoje softwaru. Navíc jeho interpretační povaha umožňuje interaktivní práci a rychlý vývoj aplikací. Tato kombinace funkcí a vlastností přispívá k jeho širokému uplatnění a popularitě. [14]

Knihovna pyModbus je výkonný nástroj pro programovací jazyk Python 3, který umožňuje implementaci a manipulaci s Modbus protokolem. Tato knihovna poskytuje robustní a snadno použitelné rozhraní pro komunikaci pomocí Modbus protokolu v prostředí jazyka Python. Podporuje jak Modbus TCP, tak Modbus RTU, což umožňuje široké spektrum aplikací v průmyslovém a automatizačním prostředí. Díky své flexibilitě a jednoduchosti je pyModbus oblíbenou volbou pro vývojáře potřebující integraci Modbus komunikace do svých projektů. [15]

Knihovna BeautifulSoup je knihovna pro programovací jazyk Python 3, která umožňuje extrakci dat z HTML a XML souborů. Tato knihovna poskytuje jednoduché a elegantní rozhraní pro analýzu webových stránek a extrakci požadovaných informací. Díky své flexibilitě a snadnému použití se stala populární volbou pro vývojáře, kteří potřebují pracovat s obsahem webových stránek. Dalším klíčovým prvkem je jednoduché použití, jelikož knihovna poskytuje intuitivní rozhraní pro navigaci v HTML stromu a vyhledávání požadovaných elementů. Taktéž podpora CSS selektorů umožňuje vyhledávání HTML elementů pomocí CSS selektorů, což poskytuje velkou flexibilitu při identifikaci požadovaných dat na webových stránkách. [16]

Knihovna Requests je jednoduchá a elegantní knihovna pro programovací jazyk Python 3, která umožňuje snadné a efektivní provádění HTTP požadavků. Poskytuje intuitivní rozhraní pro vytváření, odesílání a zpracování HTTP požadavků a odpovědí, což ji činí ideálním nástrojem pro práci s webovými API a službami. Knihovna podporuje všechny hlavní metody HTTP, automaticky spravuje cookies, a poskytuje podporu pro HTTPS a SSL/TLS, což zajišťuje bezpečnou komunikaci. [17]

2.2.4 Linux Debian

Debian je jedním z nejstarších a nejrozšířenějších operačních systémů založených na Linuxu. Jeho popularita spočívá v jeho otevřenosti, stabilitě a široké podpoře komunity. Debian poskytuje robustní a bezpečný operační systém, který je ideální pro široké spektrum použití, od osobních počítačů a serverů až po vestavěné systémy. Je známý svou stabilitou a spolehlivostí, což je důležité pro nasazení na servery a kritické systémy. Díky pečlivému testování a kontrole kvality je Debian považován za jeden z nejstabilnějších operačních systémů. Také poskytuje rozsáhlý repozitář obsahující tisíce balíčků, které lze snadno nainstalovat a spravovat pomocí nástroje APT. Tato široká nabídka software zahrnuje jak základní nástroje pro systémové administrátory, tak i aplikace pro běžné uživatele. [18]

Pro přístup na PLC z linuxové konzole bude využito programu PuTTY, PuTTY je open-source program pro různé platformy, který umožňuje uživatelům připojovat se k vzdáleným počítačům pomocí protokolů jako SSH, Telnet a sériové. Je populární mezi administrátory a vývojáři pro správu a vzdálenou správu serverů a zařízení díky jednoduchosti, spolehlivosti a rozsáhlé funkčnosti. PuTTY je volně dostupný pro stahování a použití a je oblíbeným nástrojem v oblasti IT a síťového inženýrství. [19]

Pro přenos scriptů na PLC bude využita konzolová aplikace PSCP (PuTTY Secure Copy Protocol). PSCP je určeno k přenosu souborů mezi dvěma počítači pomocí protokolu SSH. Je součástí softwarového balíčku PuTTY. [20]

Pro automatické spouštění scriptů na PLC bude použit nástroj Cron. Je to systém plánování úkolů v Unixových operačních systémech, který umožňuje automatické spouštění předem definovaných úloh v určený čas. Uživatelé mohou pomocí konfiguračního souboru crontab definovat časové plány pro své úkoly, což umožňuje automatizaci opakujících se úkolů. [21]

II. PRAKTICKÁ ČÁST

3 NÁVRH STRUKTURY SYSTÉMU PRO SPRÁVU A ŘÍZENÍ FOTOVOLTAICKÝCH A VELKOKAPACITNÍCH BATERIÍ

Hlavním komponentem celé aplikace je programovatelný logický řadič (PLC), který slouží k integraci dat a řízení připojených fotovoltaických elektráren (FVE) a velkokapacitních uložišť (BESS). Toto hlavní PLC, označené v příložené topologii jako "MAIN PLC", zajišťuje komplexní logiku fungování aplikace. PLC je založeno na operačním systému Linux, konkrétně na distribuci Linux Debian 10. Nad tímto operačním systémem je nasazeno vývojové prostředí Mervis IDE, které je dodáváno výrobcem PLC, společností UniPi. Mervis IDE poskytuje uživatelsky přívětivé prostředí pro vývoj a konfiguraci aplikací pro PLC na základě jeho specifikací a požadavků systému.

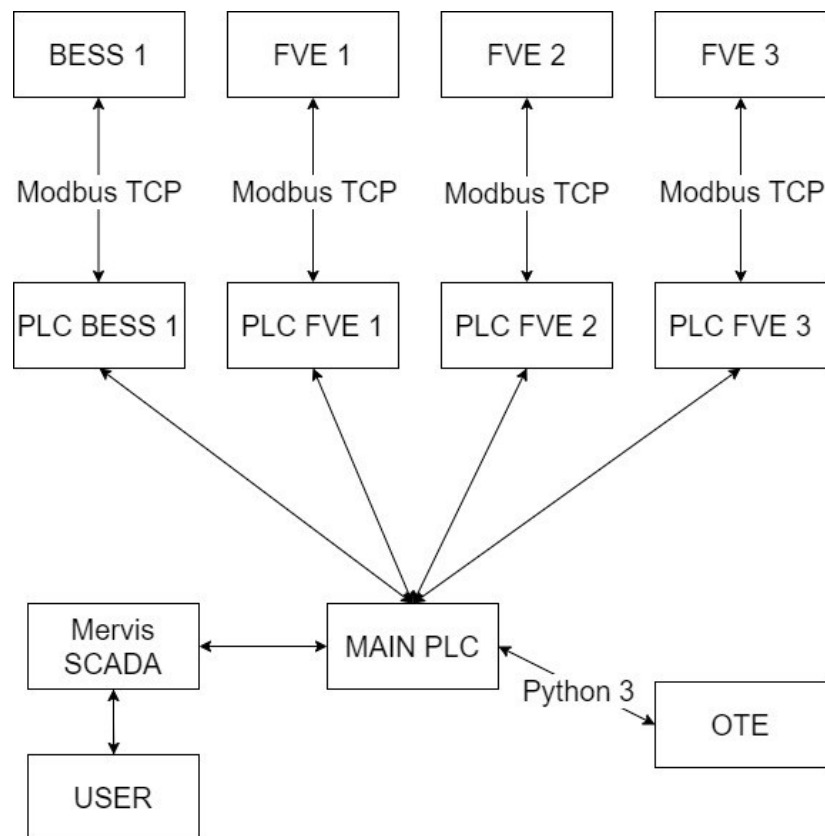
Získávání dat z portálu OTE je realizováno pomocí Python 3 skriptu, který využívá knihoven *bs4* a *requests*. Tento skript je navržen tak, aby periodicky, jednou denně, získával aktuální ceny výkupu elektrické energie po jednotlivých hodinách ze stránky portálu OTE. Získané hodnoty jsou následně zaznamenány do Modbus serveru pomocí knihovny *pymodbus* v rámci připraveného Modbus serveru v prostředí Mervis IDE. Následně jsou tyto ceny analyzovány a porovnány. Pokud jsou hodnoty kladné, což indikuje běžný stav, není nutná žádná akce. Avšak v případě, že jsou ceny záporné, což signalizuje potenciální finanční penalizaci, je spuštěna akce regulace připojených fotovoltaických elektráren (FVE). Tento proces je realizován pomocí aplikace, která interpretuje data a provádí potřebné řídicí akce v souladu s definovanými pravidly.

Na "MAIN PLC" je rovněž implementována nadstavba od společnosti UniPi, konkrétně Mervis SCADA. Tato nadstavba poskytuje dohledové centrum, které je klíčové pro monitorování a řízení fotovoltaických elektráren (FVE) a velkokapacitních baterií (BESS). Mimo plně autonomní regulace na základě vytvořeného modelu umožňuje aplikace také manuální zásahy do regulace. Tímto způsobem je zajištěna efektivní správa a optimalizace výroby a distribuce elektrické energie.

Pro navázání komunikace mezi hlavním programovatelným logickým řadičem (PLC) označeným jako "MAIN PLC" a dalšími PLC umístěnými na lokalitách, jako jsou velkokapacitní baterie (BESS) či fotovoltaické elektrárny (FVE), je využito rozhraní Shark Slave Communication Protocol (SSCP). Toto řešení umožňuje bezpečnou a spolehlivou komunikaci mezi různými částmi distribuovaného systému, přičemž data jsou přenášena prostřednictvím šifrovaného spojení v rámci izolovaného virtuálního prostoru. Díky tomu je zajištěna ochrana

dat a integrity komunikace mezi jednotlivými prvky systému, což je klíčovým aspektem pro bezpečný a efektivní provoz energetických zařízení.

Pro sběr dat a následné řízení mezi programovatelnými logickými řadiči (PLC) na konkrétní lokalitě a samotnými velkokapacitními bateriemi (BESS) či fotovoltaickými elektrárnami (FVE) je použit komunikační protokol Modbus TCP. Tento protokol, který se stal standardem v průmyslu, umožňuje spolehlivou a efektivní komunikaci mezi různými zařízeními a systémy v distribuovaném prostředí. Modbus TCP poskytuje strukturované a standardizované prostředí pro výměnu dat, což usnadňuje integraci a interoperabilitu mezi různými výrobními zařízeními a řídicími systémy. Tímto způsobem je zajištěna koordinace a synchronizace provozu mezi PLC a BESS či FVE. Podrobný vizuální přehled celé topologie zapojení a komunikace je uveden na Obrázku 18.



Obrázek 18. Topologie aplikace

4 POSTUP VÝVOJE APLIKACE PRO MONITORING A ŘÍZENÍ

4.1 Příprava vývojového prostředí

Vývoj aplikace začíná zakládáním projektu v Mervis IDE. Po spuštění Mervis IDE vytváříme novou sestavu v záložce „Soubor“. Sestavu pojmenujeme, v našem případě jako „Aplikace_BP“, a otevřeme ji v „Plném modu“, který poskytuje plnou funkčnost a možnosti programování v Mervis IDE. Následuje přidání PLC do sestavy. Fyzicky se připojíme k PLC pomocí ethernetového kabelu a vyhledáním PLC přes pravé tlačítko v sekci „Systém“ a „Přidat regulátor“ jej přidáme do projektu.

Nedílnou součástí funkční aplikace je spustitelný program v PLC. Nejprve vytvoříme projekt přes záložku „Spustitelné projekty“ a pojmenujeme jej, například „main_projekt“. V tomto projektu můžeme vytvořit program kliknutím pravým tlačítkem na „main_projekt“ a volbou „Přidat program“. Program nazveme „main“ a programovací jazyk nastavíme na FBD (Funkční blokový diagram). Program poté mapujeme na PLC dvojklikem na „PLC“, který otevře nabídku přidání tasku. Do tasku přidáme program „main“. Následuje nahrání programu do PLC pomocí možnosti „Nahrát sestavu“ v horní liště.

Pro správný provoz PLC je nezbytné ještě jej zalicencovat. Licenční klíč obdržíme od výrobce při zakoupení PLC, a to včetně Mervis runtime licence, kterou aktivujeme kliknutím pravým tlačítkem na PLC a vybráním „Operace s PLC“ a „Licencování“.

Nyní můžeme nakonfigurovat vlastnosti PLC, jako jsou síťová nastavení, parametry Mervis DB, parametry Proxy, parametry SSCP, definice uživatelů a Unipi specific. Síťová nastavení můžeme konfigurovat ručně podle rozsahu sítě nebo využít funkce DHCP. Zásadní je aby PLC mělo přístup k internetu. Parametry Mervis DB a Proxy nastavujeme povolením na „True“ a vkládáme příslušné údaje, které získáme od výrobce po zalicencování. Definice uživatelů zahrnuje uživatelská jména a hesla s dostatečně sofistikovanými kombinacemi pro zajištění bezpečnosti. V Unipi specific povolíme SSH připojení.

Po nakonfigurování vlastností můžeme tuto konfiguraci nahrát do PLC. To provedeme kliknutím pravým tlačítkem na PLC, otevřením nabídky „Operace s PLC“, volbou „Nastavení PLC“ a kliknutím na „Nahrát konfiguraci do zařízení“. Nyní máme všechny potřebné kroky pro správný provoz PLC dokončeny.

Výše uvedeným postupem provedeme vytvoření projektu pro PLC, umístěného na lokaci fotovoltaické elektrárny nebo velkokapacitního bateriového uložení. Po úspěšném vytvoření

projektu následuje fáze vytvoření komunikačního rozhraní mezi PLC a fotovoltaickými střídači či velkokapacitními bateriemi. V dnešní době podporuje každé z těchto zařízení komunikační protokol Modbus TCP nebo RTU. V našem případě se zaměříme na využití Modbus TCP.

Pro vytvoření komunikačního kanálu mezi PLC a zařízeními začínáme pravým kliknutím na „PLC“ a volbou „Přidat kanál“. Tímto krokem vytvoříme kanál, který pojmenujeme jako „FVE_Modbus“, a nastavíme komunikační protokol na Modbus s linkovým protokolem TCP.

Následně přidáme konkrétní zařízení do kanálu. Tuto operaci provedeme pravým kliknutím na kanál „FVE_Modbus“ a volbou „Přidat zařízení“. Vytvoří se nové zařízení, které pojmenujeme například jako „INV_1“, a nastavíme jeho adresu a port podle specifikací uvedených v dokumentaci výrobce střídače. IP adresu nastavíme na hodnotu, kterou má zařízení již přiřazenou v síti. Je nutné zajistit, aby jak PLC, tak i připojená zařízení, byly v rámci jednoho rozsahu sítě, což je nezbytné pro správnou komunikaci mezi nimi.

Nyní se zaměříme na přípravu Modbus registrů, které budeme číst a prostřednictvím nich regulovat střídač. Prvním krokem je zajištění Modbus registrů od výrobce, v našem případě od společnosti Sungrow. (viz Příloha I) Zde jsou uvedeny potřebné informace, jako je port a Modbus ID, které budeme vkládat do nastavení komunikace. Dále jsou uvedeny názvy, adresy, datové typy a jednotky registrů, a informace o tom, jakým Modbus funkčním kódem jsou data čtena nebo zapisována do registru, spolu s popisem datových typů a příkladem, jak sestavit Modbus dotaz na data.

Po důkladném prozkoumání dokumentace můžeme přistoupit k vytvoření skupin pro čtení nebo zápis do registrů. Dvojitým kliknutím na „INV_1“ se dostaneme do okna, kde přidáváme skupiny a datové body. Pravým kliknutím kamkoliv do okna se zobrazí dialogové okno, kde vybereme možnost „Přidat skupinu“, čímž vytvoříme novou skupinu. Jako první si zvolíme název skupiny, například „AC_power_skupina“. Typ skupiny zůstane na „ReadOnly“, protože budeme pouze číst data. Počáteční prvek čtení nastavíme podle dokumentace výrobce, v našem případě na registr číslo 36, který má Modbus adresu 8070-8073. Funkci čtení nastavíme na „F04 Read Input Register“ a počet prvků na 4. [22]

V daný moment máme vytvořenou skupinu čtení a potřebujeme do ní přidat datový bod. To provedeme podobným způsobem jako při přidání skupiny, ale tentokrát vybereme možnost

„Přidat datový bod“. Datový bod pojmenujeme například „AC_power“ a přidáme jej do skupiny „AC_power_skupina“.

Nyní se musíme zaměřit na nastavení datového bodu. Nejdříve změníme typ mapované komunikační hodnoty z „Bit“ na „BuiltIn“, což nám umožní změnit typ ST. Následně vybereme odpovídající datový typ dle dokumentace, v našem případě „U64“ neboli unsigned long integer. Poté povolíme možnost Autogen, která umožní pracovat s registrem ve vývojovém prostředí Mervis IDE.

V tuto chvíli musíme nastavit parametry datového bodu Modbus. Základním integerovým registrem Modbusu je 16 bitů, ale protože čteme registr o velikosti 64 bitů, musíme nastavit délku MultiByte (Parser) na 8 bajtů. Pořadí MultiByte zachováme jako „12345678“, což znamená čtení od nejméně signifikantního bitu (LSb) po nejvíce signifikantní bit (MSb). Tím máme vytvořenou skupinu pro čtení registru „AC_power“. Pro přidání dalších registrů pro čtení postupujeme stejně, pouze měníme nastavení adres a datových typů Modbus dle dokumentace výrobce. [22]

Pro zapisování dat do registru je postup velice podobný. Dle předchozího postupu si vytvoříme skupinu. Tu si pojmenujeme „AC_power_regulation_%_skupina“. Typ skupiny změníme na „WriteOnly“ jelikož do skupiny budeme zapisovat. Jako počáteční prvek se odkážeme opět na dokumentaci výrobce.

V tomto okamžiku chceme zapisovat do střídače regulaci v procentech výkonu střídače. Tento registr se dle dokumentace nachází na adrese 8005-8006. Zapišeme tedy 8005. Funkci zápisu nastavíme dle dokumentace na „F06 Write Single Register“. Počet prvků nastavíme na 2. Volitelná je nyní možnost zápis pouze při změně, kterou si nastavíme na „True“ z důvodu že instrukci do střídače potřebujeme zapsat pouze když se hodnota zápisu změní. Tím šetříme datovou linku a výpočetní výkon střídače. [22]

Aktuálně nás čeká přidat datový bod. Ten si přidáme dle předchozího postupu při vytváření čtecího registru. Datový bod si pojmenujeme „AC_power_regulation_%“. Přiřadíme si ho do skupiny AC_power_regulation_%_skupina“. Typ mapované kom. hodnoty změníme na „BuiltIn“ a Typ ST na „uint“ dle dokumentace a povolíme Autogen. Nyní nastavíme parametry datového bodu Modbus. Mezera offsetu pro zápis se nastaví na 4 z důvodu, že musí platit, že mezera offsetu pro zápis = offset dat + délka multibyte. Délku MultiByte tedy nastavíme dle velikosti datového bodu na 4. Pořadí MultiByte nastavíme dle dokumentace na 3412. Jedná se o takzvaný little-endian byte swap. Tím jsou všechny parametry nastaveny a

registr je možné používat v prostředí Mervis IDE. Obrázek 19 ilustruje, jak by měly vypadat nastavené vlastnosti datového bodu. [22]

Vlastnosti	
Vlastnosti datového bodu	
Název	AC_power_regulation_%
DataPoint.SortOrder	0
Interval čtení / zápis	0ms
Je specifický box	True
Přiřazené HW vlastnosti	0 items
Skupina	AC_power_regulation_%_skupina
Typ skupiny	WriteOnly
Typ mapované kom. hodnoty	BuiltIn
Typ ST	udint
Transformace	identity
Pozn.	
·	
Pozn.	
· Autogen	
Povolit Autogen	True
Cílový projekt (SWAutogen)	main_projekt
Generovaná proměnná (SW Autogen)	hw.\$INV_1_AC_power_regulation_%%\$
Generované jméno proměnné (SWAutogen)	
Retain (SWAutogen)	False
· Mapování	
IO => ST	
ST => IO	hw.\$INV_1_AC_power_regulation_%%\$
· Parametry dat. bodu Modbus	
DataPoint.Modbus.SimpleAddressString	Elem 0, Count 2
Mezera offsetu pro zápis	4
Offset dat (Parser)	0
Bitový offset (Parser)	0
Délka MultiByte (Parser)	4
Pořadí MultiByte (Parser)	3412
Pozn.	

Obrázek 19. Vlastnosti datového bodu „AC_power_regulation_%%“

Podobně jako jsme připravovali prostředí pro čtení dat ze střídačů, pustíme se do vytvoření Modbus serveru pro zápis aktuální ceny elektrické energie. Prvním krokem bude přidání nového kanálu k našemu PLC, což uděláme kliknutím pravým tlačítkem na PLC a výběrem „Přidat kanál serveru“. Takto vzniklý kanál si pojmenujeme třeba „Modbus_Server“ a nastavíme jeho protokol na „ModbusServer“ a pro komunikaci použijeme TCP.

Nyní je čas na přidání serverového zařízení. To uděláme opět pravým kliknutím, tentokrát na náš nový kanál „Modbus_Server“, kde zvolíme „Přidat zařízení serveru“. Tohle zařízení

si můžeme pojmenovat „server“. V nastavení mu dáme adresu „1“. Důležité je, aby v nastaveních TCP mělo zařízení serveru stejnou IP adresu, jakou má naše PLC v síti, a port nastavíme na 503. A s tímto krokem máme vše připraveno. Náš Modbus server je nyní schopen zapisovat aktuální cenu elektrické energie.

Jako poslední věc přípravy prostředí, kterou musíme udělat je navázat propojení mezi PLC na instalacích a hlavním PLC. Toho se dá docílit dvěma způsoby první z nich je na každém PLC na instalaci vytvořit Modbus server, kde namapujeme pouze registry, které chceme přenášet na hlavní PLC. Na hlavním PLC poté stačí vytvořit další komunikační kanály ze kterých pak jednotlivé registry budeme vyčítat a následně s nimi pracovat.

Jedním z důležitých prvků takového systému je požadavek na stejný rozsah sítě jednotlivých propojených zařízení, aby komunikace mohla probíhat. Tento požadavek jde vyřešit virtuální privátní sítí. Vzniká zde ale nové bezpečnostní riziko z hlediska zabezpečení virtuální privátní sítě.

Z toho důvodu využijeme druhý způsob a to je SSCP (shark slave communication protocol) připojení mezi jednotlivými PLC na instalacích a hlavním PLC. To provedeme tak, že na hlavním PLC klikneme pravým tlačítkem myši na „Systém“ a přidáme „Přidat zdroj dat“. Vytvoří se nám „Data Source“. Náš Data Source by se dal pojmenovat jako zdroj dat z PLC na instalaci, proto si ho přejmenujeme například na „FVE_1“. Teď musíme vybrat cestu ke spustitelné bitové kopii námi přidávaného PLC na instalaci. V průzkumníku si tedy najdeme projekt z naší instalace v podsložce „bin“ a vybereme soubor typu .exs. Aktuálně je potřeba nastavit parametry SSCP. Cílovou adresu nastavíme na „proxy.unipi.technology“, TCP port na „12348“, SSCP adresu dle nastavené adresy v projektu PLC z instalace (v našem případě 1) a Proxy ID, které získáme taktéž z projektu, který jsem vytvářeli pro PLC na instalaci. V poslední fázi implementace je vyžadováno zřízení kanálu pro SSCP připojení. Tento proces se nápadně podobá postupu přidávání Modbus kanálu, o kterém bylo pojednáno v předcházející kapitole. Pro vytvoření nového kanálu se na PLC klikne pravým tlačítkem myši, vybere se možnost pro přidání kanálu, který následně pojmenujeme „SSCP“. Jako protokol zvolíme SSCP, zatímco ostatní nastavení ponecháme v jejich výchozím stavu.

Dále přidáme k vybranému kanálu „SSCP“ nové zařízení, opět použitím pravého tlačítka myši, a toto zařízení pojmenujeme „FVE_1“. V dalším kroku je nutné specifikovat zdrojovou podstanici, pro kterou zvolíme opět „FVE_1“. Adresu zařízení nastavíme na hodnotu „1“, uživatelské jméno definujeme jako „admin“, a heslo, které bylo vybráno při vytváření

projektu pro danou lokalitu, kde je PLC instalováno. Závěrem je potřeba určit Proxy ID, které se rovněž získává z projektu PLC umístěného na příslušné lokalitě.

Tímto způsobem můžeme přidat libovolný počet PLC na instalacích do hlavního PLC ve kterém budeme s daty z těchto PLC pracovat. Jedná se jak o fyzické vstupní a výstupní porty tak i všechny vytvořené komunikační kanály v PLC na instalaci.

4.2 Vytvoření modulu pro získání ceny elektrické energie

Pro získání aktuálních cen elektrické energie jsem se rozhodl využít programovací jazyk Python 3. Navrhl jsem jednoduchý skript v Pythonu 3 (viz Příloha II), který umožňuje extrahovat aktuální hodinovou cenu elektrické energie ze stránek společnosti OTE. Pro implementaci tohoto skriptu jsem jako první importoval několik klíčových knihoven, konkrétně *requests*, *bs4*, *datetime*, *pymodbus* a *time*.

Inicializoval jsem několik konstantních proměnných a to `DATE_CZ` pro formátování času, `HOURL_NOW` pro získání aktuální hodiny a `SERVER_IP_ADDRESS` a `SERVER_PORT` pro následné ukládání hodnot na Modbus server pro další použití v Mervis IDE. [P II][23]

Samotný skript obsahuje jednu hlavní funkci jménem „`get_data_CZ`“, která má jako parametr „`date_cz`“, což je datum, pro který chceme získat data. Tato funkce sestavuje URL adresu, pomocí které se přistupuje k datům o elektrické energii na webových stránkách společnosti OTE. Datum je součástí této URL adresy. [23]

Dále funkce získává obsah stránky pomocí knihovny *requests* a analyzuje ho pomocí knihovny *bs4* pro snadnější manipulaci s HTML. Následně identifikuje konkrétní tabulku na stránce, ze které se mají získat data o elektrické energii. [23]

Funkce prochází prvních 24 řádků této tabulky a pro každý řádek získává hodnotu z prvního sloupce, který obsahuje hodinu, a používá ji jako klíč ve slovníku. Hodnota z druhého sloupce, který obsahuje cenu elektrické energie, je pak uložena jako hodnota tohoto klíče v slovníku. Na závěr funkce vrací tento slovník, který obsahuje data o elektrické energii pro dané datum, získaná ze stránky. Pro zvýšení odolnosti skriptu při selhání získání dat jsou implementovány opakované pokusy vyčítání, s možností přerušení pokud data nejsou dostupná. [23]

Pro účely zápisu dat na server je využita funkce "`write_data`", která přijímá dva parametry: "`server_ip`" a "`port`". Tyto parametry definují IP adresu a port serveru, na který mají být data zapsána. V rámci této funkce je nejprve vytvořena instance klienta protokolu Modbus TCP.

Tato instance je inicializována s poskytnutou IP adresou a portem serveru, což umožňuje klientovi navázat spojení se serverem. [23]

Po úspěšném navázání spojení mezi klientem a serverem se provede samotný zápis dat na server. Voláním funkce „get_data_CZ“, získá funkce aktuální data. Pro zápis je využita funkce "write_register", která slouží k zápisu hodnoty na konkrétní registr na serveru. Tato operace je ošetřena proti potenciálním chybám s výjimkami a problémy při spojení, což zvyšuje spolehlivost skriptu. [23]

Jako další postup nás čeká přenesení souboru „main.py“ do PLC, konkrétně do operačního systému Debian 10, kde bude skript spouštěn. K tomuto účelu jsem se rozhodl využít program PuTTY a jeho komponentu PSCP, díky které je možné soubor bezpečně přenést mezi lokálním Windows systémem a vzdáleným Linuxovým systémem. Provedení tohoto kroku začíná otevřením příkazové řádky Windows, kde jsem navigoval do složky obsahující skript „main.py“. Následně byl využit příkaz: `pscp main.py unipi@adresa_serveru:/cesta/k/ulozeni`

V tomto příkazu „unipi“ představuje uživatelské jméno na Debianovém systému a „adresa_serveru“ je IP adresa nebo doménové jméno tohoto serveru. „/cesta/k/ulozeni“ specifikuje, kam na systému bude skript „main.py“ uložen.

Po úspěšném přenesení souboru bylo nutné na Debianovém systému nainstalovat několik klíčových knihoven, aby bylo možné skript spustit. Pomocí SSH jsem se přihlásil na server a pomocí následujících příkazů jsem instaloval potřebné knihovny: `sudo pip3 install pymodbus beautifulsoup4 requests`. Tento příkaz zajišťuje instalaci knihoven *pymodbus*, *bs4* a *requests*, které jsou zásadní pro funkčnost skriptu main.py. V případě, že by pip3 nebyl na systému dostupný, je možné jej nainstalovat pomocí příkazu `sudo apt-get install python3-pip`. [24]

Dále bylo potřeba zajistit, aby se skript „main.py“ automaticky spouštěl v pravidelných intervalech. K tomu jsem využil cron, což je démon plánování úloh v Linuxu. Otevřením crontab souboru příkazem `crontab -e` jsem přidal následující řádek: `0 * * * * /usr/bin/python3 /cesta/k/main.py`. Tento řádek v crontabu nastavuje, že skript „main.py“ bude spouštěn každou hodinu na začátku hodiny. `/cesta/k/main.py` je přitom cesta k uložení skriptu v systému. [24]

Takto jsem zavedl mechanismus, který zaručuje pravidelné spouštění skriptu main.py na operačním systému Debian 10. Díky této konfiguraci bude skript automaticky každou hodinu provádět zápis dat na Modbus server.

4.3 Vývoj monitoringu a řízení

V první fázi vývoje systému je nezbytné vytvořit registr, který bude sloužit k zaznamenávání aktuálních cen výkupu elektrické energie. Tento úkol vyžaduje definici globální proměnné v programovém prostředí. K tomuto účelu je použit programový modul s názvem „main“. Pro definici globální proměnné je nejprve nutné přistoupit k uživatelskému rozhraní programu, konkrétně k levé části obrazovky, a následně aktivovat dialogové okno pro vytvoření proměnné pomocí klávesové zkratky CTRL+G. V dialogovém okně je pak možné specifikovat název proměnné, v tomto případě „cena_cz_vstup“, datový typ, kterým bude celočíselný typ („int“), počáteční hodnotu a další příznaky.

Následně je třeba tuto globální proměnnou integrovat do Modbus serveru. To se provádí rozšířením konfigurace serveru o nový registr, který bude asociován s výše definovanou proměnnou. Proces přidání registru zahrnuje výběr počátečního prvku, nastavení transformace, funkce a mapování na proměnnou „cena_cz_vstup“. Tímto způsobem je zajištěno, že registr bude pravidelně aktualizován aktuální cenou výkupu elektrické energie. Aktualizace probíhá automaticky každou hodinu prostřednictvím spuštění skriptu v jazyce Python 3 na operačním systému Debian 10.

Pokračuji v rozvoji systému tím, že přecházím k fázi získávání a vizualizace dat z fotovoltaických elektráren a velkokapacitních baterií. V rámci demonstrace aplikace byly do již existující konfigurace datových zdrojů a SSCP kanálu, o kterých byla řeč v předchozí kapitole, integrovány další komponenty: fotovoltaická elektrárna od společností SolarEdge, reprezentované v aplikaci pod názvem datového zdroje „FVE_2“, a velkokapacitní baterie od společnosti AlphaESS pod názvem datového zdroje „BESS_1“. Klíčovým zájmem v rámci dohledového centra jsou hodnoty činného a jalového výkonu jak pro fotovoltaické elektrárny, tak pro bateriová úložiště. K těmto ukazatelům přibývá u velkokapacitních baterií také sledování aktuálního stavu nabití baterie.

Důležitou součástí správy těchto zařízení je implementace funkcionalit pro jejich vypínání a zapínání, což přispívá k flexibilnímu řízení provozu. V případě velkokapacitních baterií je zvláště podstatná možnost řízení procesů nabíjení a vybíjení.

V dalším kroku implementace systému je nezbytné rozšířit jeho funkcionalitu o proměnné, které umožňují čtení dat ze střídačů instalovaných na různých lokalitách. Tato data jsou přenášena prostřednictvím SSCP protokolu do hlavního PLC, kde jsou následně zpracovávána.

Pro ilustraci tohoto procesu využijeme data zaregistrovaná pod „FVE_1“, která již obsahují předdefinované registry pro čtení údajů ze střídačů vyráběných společností Sungrow.

K přidání nových proměnných je nutné v uživatelském rozhraní otevřít kanál SSCP „FVE_1“ a prostřednictvím kontextového menu (pravé tlačítko myši) přidat novou skupinu, kterou pojmenujeme „cteni_skupina“. Veškeré další nastavení skupiny ponecháme v jejich základních hodnotách. Následně do této skupiny přidáme nový datový bod, který pro účely demonstrace pojmenujeme „AC_Power“. V nastavení tohoto datového bodu přiřadíme bod do skupiny „cteni_skupina“ a aktivujeme možnost Autogen, což umožní využití datového bodu v programu.

Pro výběr zdrojové proměnné využijeme spojení SSCP, díky čemuž získáme přehled všech proměnných dostupných z PLC na dané lokalitě. Vzhledem k záměru monitorovat hodnotu AC výkonu vybereme proměnnou „FVE_1_AC_power“, která byla předem definována v rámci přípravy prostředí. Datový typ této proměnné se automaticky nastaví na ulint, což odpovídá formátu proměnné AC_power v lokálním PLC. Díky této konfiguraci můžeme s proměnnou „FVE_1_AC_Power“ efektivně pracovat v rámci programu na hlavním PLC.

Proces přidání proměnných pro zápis se neliší od výše popsaného postupu, s výjimkou změny typu skupiny z „ReadOnly“ na „WriteOnly“. Tento krok je zásadní pro správnou konfiguraci systému, umožňující nejen čtení, ale i zápis dat.

Při implementaci systému pro monitorování a řízení zařízení je nezbytné zohlednit specifika komunikace s různými typy zařízení, zejména když jde o rozdíly v protokolu Modbus, který je široce využíván pro vyčítání dat a ovládání střídačů. Různí výrobci nabízejí odlišné přístupy k organizaci a přístupu k datům, což má zásadní vliv na způsob integrace a zpracování informací v rámci dohledového systému.

Společnost Sungrow, například, poskytuje možnost agregace dat o celkovém výkonu všech střídačů v rámci jedné instalace do jediného registru. Na druhé straně, firma SolarEdge (viz Příloha III) zvolila metodu, při které se činný výkon každého střídače na instalaci vyčítá zvlášť, avšak vyžaduje dodatečný krok v podobě přepočítávání dat pomocí scale faktoru. Pokud jde o velkokapacitní baterie od společnosti AlphaESS (viz Příloha IV), proces ovládání zahrnuje specifický postup, kde je nejprve nutné převzít řízení nad baterií, než začne reagovat na příkazy.

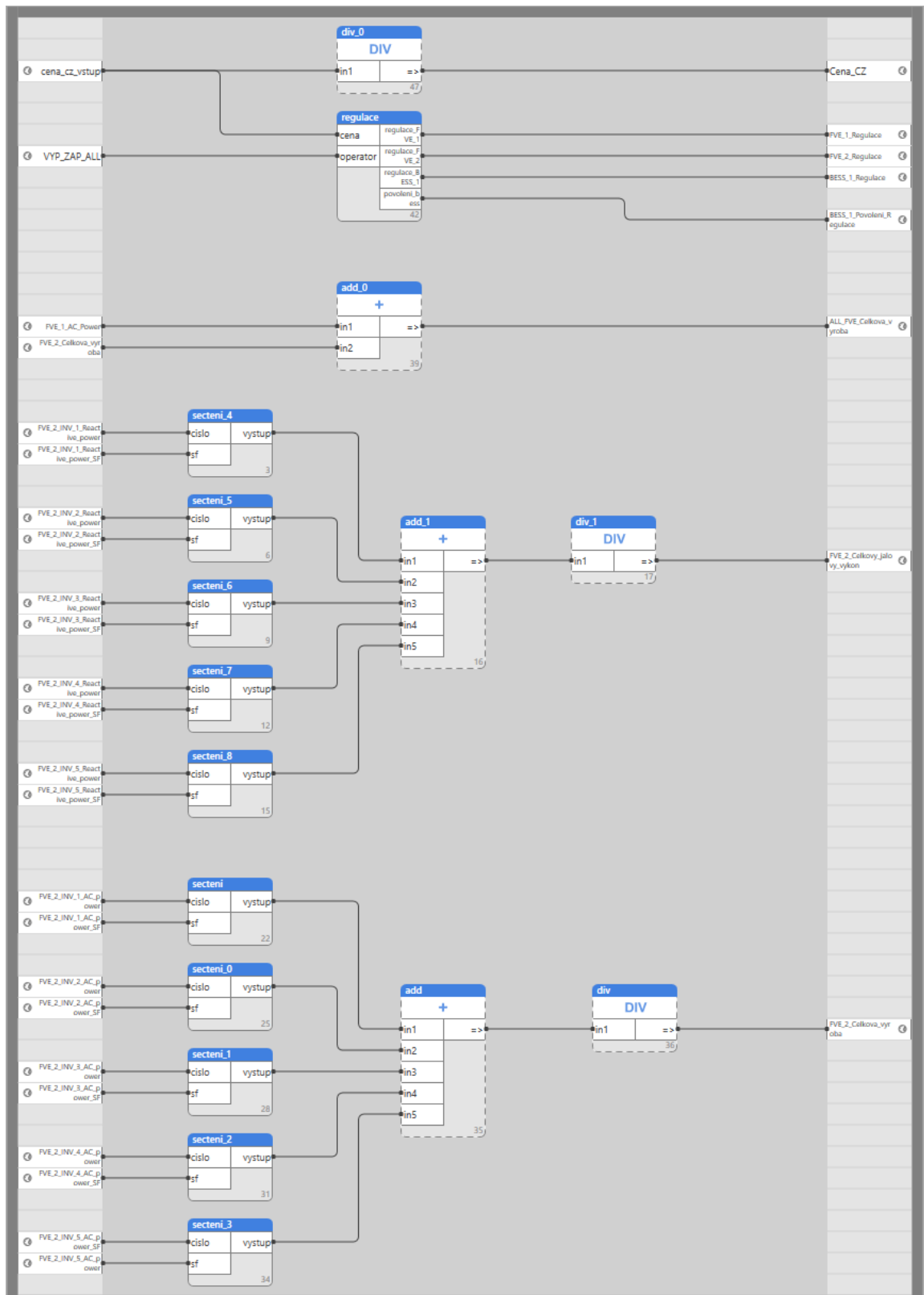
Po úspěšné integraci a sběru všech nezbytných dat z různých střídačů je dalším krokem transformace těchto dat do formátu vhodného pro zobrazování. Tento proces zahrnuje

konverzi surových dat do uživatelsky přívětivých hodnot, které přesně reflektují aktuální stav systému, a jsou snadno interpretovatelné pro operátory dohledového centra.

Provedení této transformace vyžaduje aplikaci specifických převodních koeficientů, scale faktorů a dalších výpočetních operací, které přizpůsobí data formátu požadovanému pro monitorovací dashboardy, reporty a analytické nástroje. Příkladem může být převod výkonu z technických jednotek do jednotek běžně používaných v energetice, jako jsou kilowatty.

Pokud data pocházejí ze zařízení různých výrobců, které využívají rozličné metody pro extrakci dat, jak je to například u střídačů od společnosti SolarEdge vyžadujících aplikaci scale faktoru pro korektní interpretaci, je nezbytné navrhnout a implementovat funkční blok (FBD) pojmenovaný „secteni“. Tento blok je speciálně navržen pro automatické přepočítávání dat podle potřeby, zajišťujíc tento kritický převod dat bez nutnosti manuálního zásahu.

Ve vývoji našeho systému pro monitorování a řízení energetických zdrojů jsme dosáhli fáze, kdy aplikace již umí aktivně reagovat na změny v ceně výkupu elektrické energie. Tato schopnost umožňuje automatické vypínání připojených fotovoltaických elektráren a regulaci vybíjení velkokapacitní baterie. Funkčnost programu je navržena tak, aby operátor dohledového centra mohl zasahovat do systému ručně - jednotlivě vypínat každou FVE nebo velkokapacitní baterii, případně vypnout všechny FVE současně. Kromě toho systém umožňuje, aby se FVE automaticky vypínaly a velkokapacitní baterie přestala vybíjet v situacích, kdy tržní cena výkupu elektrické energie spadne do záporných hodnot. Veškerá tato funkcionálnita je navržena ve funkčním bloku s názvem „regulace“. Obrázek 20 znázorňuje celkový vytvořený program „main“ ve formě FBD.



Obrázek 20. Celkový pohled na vytvořený program „main“ ve FBD

Na začátku procesu vytváření SCADA aplikace je klíčové založit SCADA projekt přímo na webovém portálu výrobce PLC, v tomto případě společnosti Unipi. Tento krok nám umožní

rezervovat prostor v cloudovém úložišti Unipi, kam budeme moci nahrát naši SCADA aplikaci. Kromě toho obdržíme zásadní informace nutné pro vzájemné propojení vývojového prostředí Mervis IDE s cloudovou platformou. Mezi tyto informace patří unikátní název projektu, ID projektu a přiřazená doména, které budou sloužit jako most mezi naším lokálním vývojem a cloudovou infrastrukturou.[25]

Následujícím krokem je vytvoření tzv. terminálu v našem vývojovém prostředí, což představuje virtuální reprezentaci našeho SCADA projektu. Terminál slouží jako centrální bod, který obsahuje kompletní informace o konfiguraci projektu. Vytvoření terminálu zahrnuje výběr typu SCADA verze, kterou chceme používat - v našem případě „Mervis SCADA rev 1.0“. Po vytvoření terminálu je důležité přidat k němu specifické PLC, které bude SCADA aplikace monitorovat a ovládat. To se provádí přidáním zařízení do kanálu terminálu, což umožní následné nastavení připojení a komunikace s fyzickým PLC.

Dalším důležitým krokem je důkladné nastavení vlastností terminálu tak, aby odpovídaly informacím získaným při registraci SCADA projektu v cloudu. Je třeba zde specifikovat ID projektu, název projektu a doménu. V této fázi je také nutné konfigurovat připojení PLC k terminálu, včetně nastavení parametrů pro SSCP komunikaci, jako jsou uživatelské jméno, heslo a proxy ID, a TCP parametry jako adresa zařízení, port a šifrování SSL.

Abychom mohli ukládat data do historie, je nezbytné zřídit profil pro historické záznamy. Tento krok realizujeme kliknutím pravým tlačítkem myši na sekci „Historie“ a výběrem možnosti „Přidat novou historii“. Po vytvoření nového historického záznamu tento přejmenujeme na „History“. V nastaveních tohoto historického profilu pak vybereme, aby byla historie uchovávána přímo v PLC, což nastavíme na hodnotu True. Dále je důležité integrovat tento historický záznam do konfigurace PLC, přičemž do příslušného pole pro identifikaci runtime prostředí zaznamenáme „History“. Tímto postupem zabezpečíme, že máme možnost kdykoli si z PLC vyžádat a stáhnout historická data.

Následně přistoupíme k začlenění všech proměnných do tabulky datových bodů, které plánujeme vizualizovat v rámci SCADA aplikace. Pro tento účel otevřeme panel tabulky datových bodů a vybereme relevantní proměnné pro náš projekt. Mezi tyto proměnné patří například: Cena_CZ, VYP_ZAP_ALL (příkaz pro zapnutí/vypnutí všech zařízení), ALL_FVE_Celkova_vyroba (celková produkce všech fotovoltaických elektráren), FVE_1_AC_power (výkon první FVE), FVE_1_Reactive_power (jalový výkon první FVE),

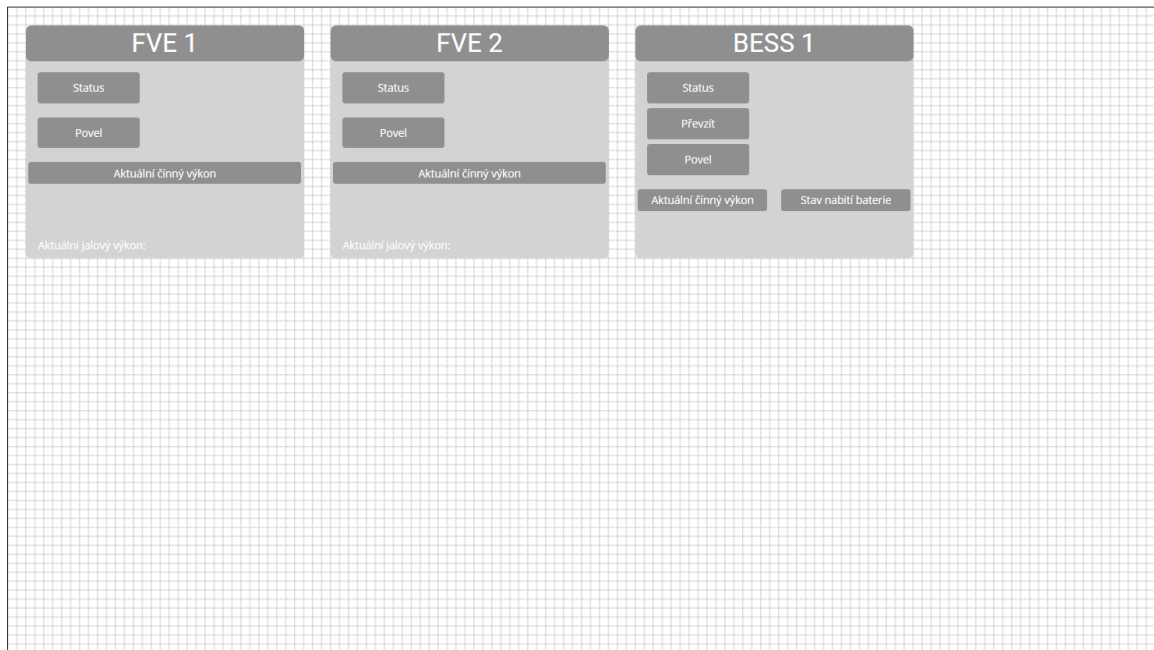
FVE_1_Regulace (regulační proměnná první FVE), a další specifické proměnné pro fotovoltaické elektrárny, velkokapacitní baterie a jejich řízení.

Po výběru všech potřebných datových bodů je dalším krokem jejich začlenění do historických záznamů, což umožní jejich sledování a analýzu v průběhu času. Tento proces začínáme označením všech přidanych datových bodů, následně pomocí kontextového menu (pravým kliknutím myši) vybereme možnost přidání těchto bodů do historie.

Dále provedeme import všech datových bodů do prostředí SCADA. Toho dosáhneme pravým kliknutím na panel „tabulka_datovych_bodu“, kde zvolíme možnost „Import SCADA proměnných“. V následujícím dialogovém okně pak vybereme volbu „Přidat (vše)“, čímž se všechny předem vybrané datové body stávají součástí našeho SCADA projektu. S tímto krokem jsou datové body úspěšně připraveny pro vizualizaci a další zpracování v rámci SCADA systému.

Nyní, když jsou datové body integrovány do SCADA, můžeme přistoupit k návrhu a implementaci grafického rozhraní dohledového centra.


V procesu vytváření grafického rozhraní SCADA dohledu využijeme různé předdefinované HMI objekty dostupné v Mervis IDE, mezi něž patří Digital Indicator, Digital Setter, Digital Setter Buttons, Rectangle, Text a Value Indicator. Začneme konstrukcí vizuálního panelu pro zobrazení dat z fotovoltaických elektráren, využijeme k tomu objekty Rectangle a Text. V tomto panelu budou prezentovány informace jako stav fotovoltaické elektrárny (zapnuto/vypnuto), povel k aktivaci či deaktivaci, aktuální činný a jalový výkon. Pro velkokapacitní baterie pak zobrazíme data o činném výkonu, stavu nabití, regulaci baterie a ovládací prvky pro manuální nabíjení nebo vybíjení. Na Obrázku 21 lze vidět, jak připravené rozhraní z HMI objektů vypadá.



Obrázek 21. Rozhraní z vizuálních HMI objektů

Pro zobrazování hodnot využijeme objekt Value Indicator, kde každé proměnné přiřadíme odpovídající formát a jednotku. Tento krok zopakujeme pro všechny analogové hodnoty, které chceme v SCADA dohledu monitorovat.

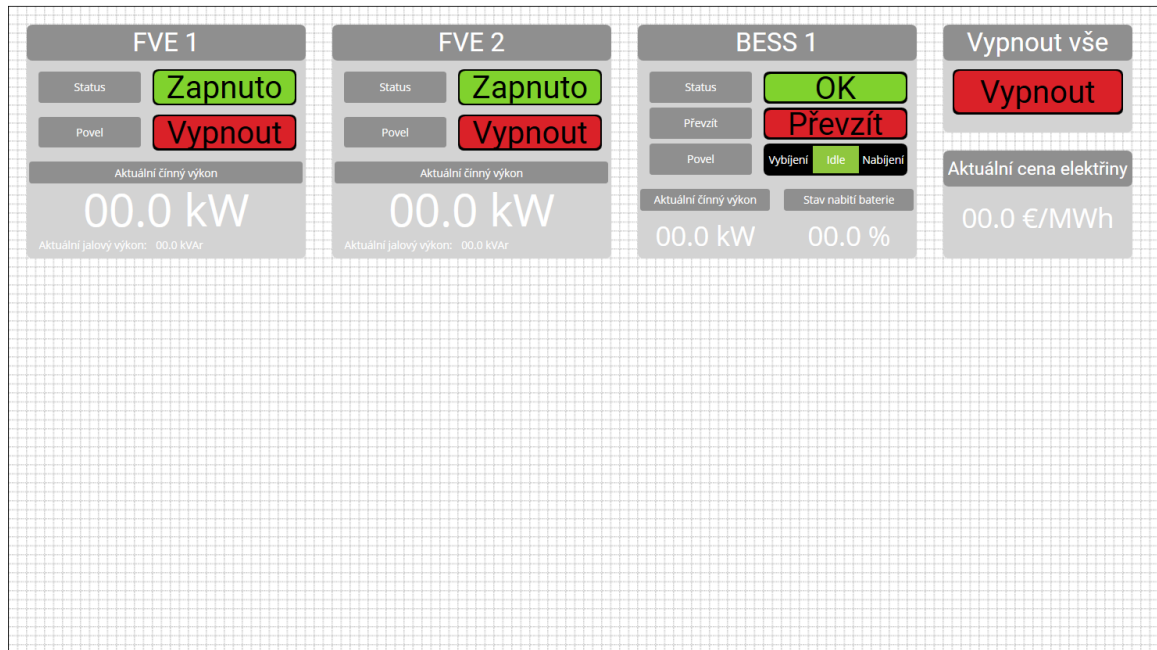
Další krok zahrnuje implementaci Digital Indicators, které budou indikovat stav zařízení. Pro každý takový indikátor přiřadíme proměnnou a definujeme stavy podle daných povelů. Na Obrázku 22 jsou zobrazeny vlastnosti HMI prvku Digital Indicator, kde je namapována proměnná včetně dvou stavů. Podobné nastavení, kromě nastavení stavů, platí i pro HMI prvek Value Indicator.

Vlastnosti	
Vlastnosti prvku	
Název	Digital Indicator
Bublinová nápověda	
Styly objektů	Default
Pozn.	
· Pozice	
Šířka	160
Výška	40
Nahoru	70
Vlevo	160
Z-index	25
· HMI	
Proměnná	PLC/hw.FVE_1_Regulace
Mapování stavů	2 items
· Text	
Popředí	<input type="text"/>
Velikost fontu	90
Písmo	Open Sans
Tučné	False
Kurzíva	False
Svislé zarovnání textu	Na střed
Vodorovné zarovnání textu	Na střed
· Vizualizace	
Pozadí	<input type="text"/>
Barva okraje	
Tloušťka okraje	2
Poloměr rohů	6
Zdroj obrázku	
Natažení obrázku	None

Obrázek 22. Vlastnosti Digital Indicator

Povelové akce zajistíme pomocí Digital Setterů, které nám umožní mapovat proměnné a definovat stavy pro spínání. Specificky u baterie je postup mírně odlišný, zde po manuálním převzetí řízení umožníme operátorovi nabíjení nebo vybíjení pomocí Digital Setter Buttons, kde si přednastavíme proměnné pro regulaci a vytvoříme tři režimy - základní stav, nabíjení a vybíjení.

K finálním prvkům patří Value Indicator, který bude zobrazovat aktuální cenu výkupu elektrické energie, a Digital Setter, umožňující operátorovi vypnout všechny fotovoltaické elektrárny či zastavit vybíjení baterie jedním tlačítkem. Na Obrázku 23 můžeme vidět konečnou verzi dohledového centra.



Obrázek 23. Konečná verze dohledového centra

Pro zpřístupnění SCADA aplikace na cloudové platformě Unipi je nezbytné celý projekt zkompileovat. Kompilace vytvoří v adresáři projektu exportní složku s komprimovaným balíčkem, který následně nahrajeme do našeho SCADA projektu na webovém portálu Unipi. Díky tomu získáme přístup k dohledovému systému odkudkoliv.[25]

5 POŽADAVKY NA SPOLEHLIVOST A ZABEZPEČENÍ APLIKACE

Jedním z klíčových požadavků na spolehlivost celé aplikace je zajištění nepřetržitého provozu hlavního PLC, který je základem pro provoz SCADA aplikace. Abychom výrazně omezili jakýkoliv výpadek, ať už internetového či elektrického připojení, doporučuje se umístění PLC do serverovny. Tato serverovna by měla být vybavena redundantními napájecími systémy, jako jsou nezávislé zdroje napájení (UPS), které zajistí kontinuitu napájení i v případě výpadku elektrické energie. Důležitá je také implementace více nezávislých síťových připojení, která zaručí nepřetržitý přístup k síti. Kromě toho je nezbytné pravidelně provádět údržbu a aktualizaci softwaru a hardwaru, aby všechny komponenty systému byly v optimálním stavu a chráněny proti známým bezpečnostním hrozbám.

Dalším doporučením pro efektivní fungování takové aplikace je decentralizace SCADA aplikací. Toto rozložení slouží jako záložní řešení, kdy na každém PLC na lokalitě vytvoříme samostatnou SCADA aplikaci s aktuálními daty a možností ovládání elektrárny. V případě výpadku hlavního PLC toto záložní řešení umožní, byť možná s vyššími nároky na ovládání, aby bylo možné i nadále řídit fotovoltaické elektrárny nebo velkokapacitní baterie z jednotlivých lokalit. Tato konfigurace poskytuje kontinuitu operací a zajišťuje, že řízení může pokračovat i v situaci, kdy hlavní systém selže.

Vzhledem k tomu, že se jedná o aplikaci generující poměrně velký zisk, je naprosto nepřijatelné, aby do ní mohla externí osoba jakkoliv zasahovat či se do ní nabourat. Proto je zásadní správně nastavit přístupová oprávnění jak pro PLC skrze Mervis IDE, tak pro webovou SCADA aplikaci. Abychom zabránili připojení externího uživatele přes Mervis IDE k PLC, je klíčové, jak již bylo zmíněno v jedné z předchozích kapitol, nastavit přístupová hesla pro různé úrovně uživatelů. Uživatel „Engineering“ má plný přístup umožňující veškerou manipulaci s PLC. Uživatel „Plné řízení“ může číst hodnoty proměnných a zapisovat do nich. Uživatel „Pouze čtení“ má možnost pouze číst proměnné. Je důležité zvolit hesla, která jsou dostatečně silná, aby nebyla snadno prolomitelná a efektivně zamezila neautorizovanému přístupu k PLC. Pokud jde o webový přístup k SCADA aplikaci, každý uživatel, který si přeje aplikaci využívat, si musí nejprve vytvořit účet. Tento účet je pak nutné manuálně přidat do patřičného projektu webové SCADA aplikace na portálu Unipi. Při přidávání účtu lze specifikovat, zda má přidáný uživatel možnost pouze číst data nebo zda mu bude umožněno i zapisovat hodnoty a ovládat aplikaci skrze webové rozhraní. Takové nastavení přístupových práv umožňuje detailní kontrolu nad tím, kdo může provádět určité operace v

rámci systému. Přístup na webovou SCADA aplikaci je možný přes stránky <https://scada.unipi.technology/>, kde je vyžadováno zadání uživatelského jména a hesla. Tento mechanismus zajišťuje, že pouze ověřeni uživatelé mohou získat přístup k aplikaci, což výrazně snižuje riziko neoprávněného přístupu a zvyšuje celkovou bezpečnost systému.

[26]

6 PŘÍNOS APLIKACE

Přínos aplikace vidím v možnosti sjednocení různých typů energetických zařízení do jednoho centralizovaného dohledového centra. Toto centrum významně usnadňuje práci operátorům tím, že konsoliduje všechny důležité informace na jednom místě, což jim šetří čas. Systém může efektivně zpracovat desítky instalací a vytvořit tak robustní dohledové centrum. Jediným předpokladem pro rozšíření systému je přítomnost internetového připojení na dané lokalitě a možnost navázání komunikace s připojovaným zařízením. Následně se na lokalitu nainstaluje předpřipravené PLC a v hlavním PLC se rozšíří dohledové centrum o novou instalaci.

Jako další přínos tohoto dohledového centra považuji možnost implementace různých automatizací. V našem případě jde o vypínání fotovoltaických elektráren při záporné ceně elektrické energie. Toto řešení ocení operátoři, když se v blízké době ceny výkupu elektrické energie již nebudou měnit v hodinových intervalech, ale v patnáctiminutových. Tato automatizace je velmi efektivní, protože operátoři již nemusí manuálně řídit fotovoltaické elektrárny, což by při přechodu na patnáctiminutové intervaly nebylo při velkém počtu instalací zvládnutelné.

Další významný přínos aplikace spočívá ve sběru a analyzování dat ze všech připojených fotovoltaických elektráren. V případě, že provozujeme dvě elektrárny v geografické blízkosti, z nichž jedna je vybavena systémem pro natáčení fotovoltaických panelů, nabízí se nám unikátní příležitost pro porovnání efektivity obou instalací. Díky shromážděným datům můžeme vyhodnotit, zda se investice do technologie natáčení panelů skutečně vyplatila, což umožňuje informovanější rozhodování o dalších investicích do našich technologií.

Aplikace je navržena s důrazem na flexibilitu a rozšiřitelnost. Díky modulární architektuře umožňuje snadné přidávání nových funkcionalit a zařízení bez potřeby zásadních zásahů do stávající struktury. Tento přístup zajišťuje, že aplikace může růst a adaptovat se na nové požadavky a technologie bez významných nákladů nebo technických obtíží. Je tedy plně připravena na implementaci veškerých možných rozšíření, což ji činí vysoce udržitelnou a budoucností orientovanou řešením. Tímto způsobem aplikace nejen že zvyšuje efektivitu a snižuje operativní náklady, ale také zůstává v souladu s neustále se vyvíjejícím trhem energetických technologií.

V kontextu rozšiřitelnosti můžeme mluvit například o implementaci vyčítání předpovědi počasí nebo pravděpodobného osvitů na lokalitách fotovoltaických elektráren. Toto rozšíření

umožňuje vyvinout sofistikované řízení i pro fotovoltaické elektrárny spojené s velkokapacitními bateriemi. Představuji si aplikaci, která porovnává tři veličiny – výkup elektrické energie, teplotu a pravděpodobný osvit – a na základě nich plánuje řízení velkokapacitní baterie. Toto může být užitečné, když víme, že v odpoledních hodinách bude vysoký osvit a ideální teplota pro výrobu fotovoltaickými panely, ale zároveň panuje záporná cena výkupu elektrické energie. V takových případech můžeme velkokapacitní baterii vybit během noci nebo dopoledne a začít ji nabíjet, jakmile se cena výkupu elektrické energie překlápí do záporných hodnot, čímž efektivně využijeme vysokou produkci fotovoltaických panelů za ideálních podmínek a zároveň neztrácíme příjmy z výroby elektrické energie, o kterou bychom jinak přišli bez využití velkokapacitní baterie.

ZÁVĚR

V rámci této bakalářské práce byl vyvinut a detailně popsán systém pro správu a řízení fotovoltaických elektráren. Hlavním cílem práce bylo vytvoření aplikace, která umožňuje centralizované dohledové řízení, efektivní sběr dat a automatizaci řídicích procesů, a tím zvyšuje efektivitu provozu fotovoltaických elektráren. Dosažení tohoto cíle bylo podpořeno implementací funkcionalit, jako je sjednocení různých energetických zařízení do jednoho dohledového centra a automatizace vypínání elektráren v reakci na tržní ceny výkupu elektrické energie.

V budoucích krocích by bylo vhodné zkoumat možnosti dalšího rozvoje aplikace, například prostřednictvím integrace pokročilých algoritmů pro optimalizaci rozhodovacích procesů a zlepšení předpovědí výkonnosti elektráren či rozložení výroby do velkokapacitní baterie na základě reálných meteorologických dat a ceny výkupu elektrické energie.

Jako nevýznamnější milník je třeba zmínit, že aplikace je již nasazena a aktivně testována v naší firmě. První zpětné vazby od operátorů potvrzují, že aplikace výrazně zjednodušuje jejich pracovní procesy, šetří čas díky automatizovanému vypínání fotovoltaických elektráren při záporných cenách výkupu elektrické energie a zároveň přispívá k dalšímu zvýšení finančních výnosů společnosti. Tyto dodatečné příjmy vznikají především díky schopnosti operátorů efektivněji vypínat fotovoltaické elektrárny a aktivně dorovnávat odchylky v distribuční síti, což má přímý dopad na zlepšení stability sítě a optimalizaci energetických nákladů. Tato pozitivní zpětná vazba zdůrazňuje úspěšnou implementaci a funkčnost systému, což potvrzuje jeho efektivitu a přínos pro celou organizaci.

Tato práce tak představuje zásadní pokrok v oblasti správy fotovoltaických elektráren a nastavuje směr pro budoucí inovace v energetickém managementu. Vzhledem k dosaženým výsledkům a potenciálu pro další rozvoj aplikace se otevírají nové možnosti pro zlepšení efektivitu a optimalizaci operací v energetickém sektoru.

SEZNAM POUŽITÉ LITERATURY

- [1] *SolarEdge Monitoring*, 2024. Online. SolarEdge Monitoring. Dostupné z: <https://monitoring.solaredge.com/solaredge-web/p/login>. [cit. 2024-02-19].
- [2] *Fronius Solar.Web*, 2024. Online. Fronius Solar.Web. Dostupné z: <https://www.solarweb.com>. [cit. 2024-02-19].
- [3] *FusionSolar*, Copyright © 2011-2024. Online. HUAWEI DIGITAL POWER TECHNOLOGIES CO., LTD. FusionSolar. Dostupné z: <https://eu5.fusionsolar.huawei.com>. [cit. 2024-02-19].
- [4] *ESS Platform*, 2024. Online. ESS Platform. Dostupné z: <https://poweroad.cn/#/>. [cit. 2024-02-19].
- [5] *ISolarCloud*, Copyright © 2024. Online. ISolarCloud. Dostupné z: <https://www.isolarcloud.eu>. [cit. 2024-02-19].
- [6] *Alpha ESS Monitoring*, 2024. Online. ALPHA ESS CO., LTD. Alpha ESS Monitoring. Dostupné z: <https://cloud.alphaess.com>. [cit. 2024-02-19].
- [7] *Fotovoltaika*, 2024. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2024, 4.1.2024. Dostupné z: <https://cs.wikipedia.org/wiki/Fotovoltaika>. [cit. 2024-02-20].
- [8] *Battery storage power station*, 2024. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2024, 20.2.2024. Dostupné z: https://en.wikipedia.org/wiki/Battery_storage_power_station. [cit. 2024-02-20].
- [9] *OTE - Základní údaje*, 2024. Online. OTE. Dostupné z: <https://www.ote-cr.cz/cs/o-spolecnosti/zakladni-udaje>. [cit. 2024-02-20].
- [10] *Unipi Patron S107*, © 2014-2024. Online. Unipi Patron S107. Dostupné z: <https://www.unipi.technology/cs/unipi-patron-s107-p370>. [cit. 2024-03-13].
- [11] ŠMEJKAL, Ladislav a MARTINÁSKOVÁ, Marie, 1999. *PLC a automatizace*. Praha: BEN - technická literatura. ISBN 80-860-5658-9.
- [12] *Mervis*, © 2014-2024. Online. Mervis. Dostupné z: <https://www.unipi.technology/cs/produkty/mervis-43>. [cit. 2024-03-13].
- [13] *Modbus FAQ: About the Protocol*, © 2005-2024. Online. MODBUS ORGANIZATION, INC. Modbus FAQ: About the Protocol. Dostupné z: <https://modbus.org/faq.php>. [cit. 2024-03-13].

- [14] *The Python Tutorial*, © 2001-2024. Online. PYTHON SOFTWARE FOUNDATION. The Python Tutorial. 16.3.2024. Dostupné z: <https://docs.python.org/3/tutorial/index.html#tutorial-index>. [cit. 2024-03-17].
- [15] *PyModbus - A Python Modbus Stack*, 2024. Online. PyModbus - A Python Modbus Stack. Dostupné z: <https://pymodbus.readthedocs.io/en/latest/source/readme.html>. [cit. 2024-03-17].
- [16] *Beautiful Soup*, © 1996-2024. Online. RICHARDSON, Leonard. Beautiful Soup. 17.1.2024. Dostupné z: <https://www.crummy.com/software/BeautifulSoup/>. [cit. 2024-03-17].
- [17] *Requests: HTTP for Humans™*, 2024. Online. Requests: HTTP for Humans™. Dostupné z: <https://requests.readthedocs.io/en/latest/>. [cit. 2024-03-17].
- [18] *Proč používat Debian*, © 1997-2024. Online. SOFTWARE IN THE PUBLIC INTEREST, INC. Proč používat Debian. 3.1.2024. Dostupné z: https://www.debian.org/intro/why_debian. [cit. 2024-03-17].
- [19] *Chapter 1: Introduction to PuTTY*, 2023. Online. Chapter 1: Introduction to PuTTY. 18.12.2023. Dostupné z: <https://the.earth.li/~sgtatham/putty/0.80/html/doc/Chapter1.html#intro>. [cit. 2024-03-19].
- [20] *Chapter 5: Using PSCP to transfer files securely*, 2023. Online. Chapter 5: Using PSCP to transfer files securely. 18.12.2023. Dostupné z: <https://the.earth.li/~sgtatham/putty/0.80/html/doc/Chapter5.html#pscp>. [cit. 2024-03-19].
- [21] *Cron*, 2024. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 27.2.2024. Dostupné z: <https://en.wikipedia.org/wiki/Cron>. [cit. 2024-03-19].
- [22] *MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3*, 2012. Online. <https://modbus.org/>. Dostupné z: https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf. [cit. 2023-11-12].
- [23] PILGRIM, Mark, c2010. *Ponořme se do Python(u) 3: Dive into Python 3*. CZ.NIC. Praha: CZ.NIC. ISBN 978-80-904248-2-1.

- [24] KAMENÍK, Pavel, 2011. *Příkazový řádek v Linuxu: praktická řešení*. Brno: Computer Press. ISBN 978-80-251-2819-0.
- [25] *Unipi*, © 2014-2024. Online. UNIPI TECHNOLOGY S.R.O. Unipi. Dostupné z: <https://www.unipi.technology>. [cit. 2024-04-04].
- [26] *Mervis SCADA*, © 2014-2024. Online. UNIPI TECHNOLOGY S.R.O. <https://scada.unipi.technology/>. Dostupné z: <https://scada.unipi.technology/>. [cit. 2024-04-19].

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

FVE	Fotovoltaická elektrárna.
PLC	Programovatelný logický automat
SCADA	Supervisory Control And Data Acquisition
ST	Strukturovaný text
FBD	Function Block Diagram
BESS	Battery Energy Storage System (Velkokapacitní baterie).
SSH	Secure Shell
SSCP	Shark Slave Communication Protocol
HTML	Hypertext Markup Language
XML	Extensible Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
SSL	Secure Sockets Layer
TLS	Transport Layer Security
API	Application Programming Interface
CSS	Cascading Style Sheets
TCP	Transmission Control Protocol
IP	Internet Protocol
URL	Uniform Resource Locator
LSb	Least Significant bit
MSb	Most Significant bit
UPS	Uninterruptible Power Supply
PSCP	PuTTY Secure Copy Protocol

SEZNAM OBRÁZKŮ

Obrázek 1. Dohledové centrum firmy SolarEdge [1]	9
Obrázek 2. Dohledové centrum firmy Fronius [2]	10
Obrázek 3. Dohledové centrum firmy Huawei [3]	11
Obrázek 4. Dohledové centrum firmy Huawei – nastavení maximálního výkonu střídače [3]	11
Obrázek 5. Dohledové centrum firmy Poweroad [4]	12
Obrázek 6. Dohledové centrum firmy Sungrow [5]	13
Obrázek 7. Dohledové centrum firmy Sungrow – sekce „Device Information“ [5] ..	14
Obrázek 8. Dohledové centrum firmy Sungrow - sekce "Curve“ [5]	15
Obrázek 9. Dohledové centrum firmy Sungrow - sekce "Fault" a chybová hlášení střídačů [5]	15
Obrázek 10. Dohledové centrum firmy Sungrow - sekce "Plant Configuration“ [5] ..	16
Obrázek 11. Dohledové centrum firmy Sungrow - sekce „Settings“ [5]	17
Obrázek 12. Dohledové centrum firmy Sungrow - Vzdálený přístup pro řízení [5] ..	18
Obrázek 13. Dohledové centrum firmy Sungrow - manuální ovládání fotovoltaických střídačů [5]	18
Obrázek 14. Dohledové centrum firmy AlphaESS [6]	19
Obrázek 15. Dohledové centrum firmy AlphaESS - sekce "Power Diagram" [6]	20
Obrázek 16. Dohledové centrum firmy AlphaESS - sekce "Running Logs" [6]	20
Obrázek 17. Dohledové centrum firmy AlphaESS – sekce „System Setup“ [6]	21
Obrázek 18. Topologie aplikace	29
Obrázek 19. Vlastnosti datového bodu „AC_power_regulation_%“	33
Obrázek 20. Celkový pohled na vytvořený program „main“ ve FBD	40
Obrázek 21. Rozhraní z vizuálních HMI objektů	43
Obrázek 22. Vlastnosti Digital Indicator	44
Obrázek 23. Konečná verze dohledového centra	45

SEZNAM PŘÍLOH

- P I: Modbus datasheet Sungrow
- P II: Python script
- P III: Modbus registry SolarEdge
- P IV: Modbus registry AlphaESS

PŘÍLOHA P I: MODBUS DATASHEET SUNGROW



Classification: Public
Clean power for all

Logger Communication Protocol

AW0 1.0.2.5

For models:

Logger1000

Logger3000

I. General Description

This protocol applies to the communication between the Logger3000/Logger1000 and the host computer monitoring software. It adopts MODBUS RTU/TCP communication protocol. This protocol can read the run information of the Logger3000/Logger1000 and the sensor information configured for the Logger3000/Logger1000 in real time. The Logger3000/Logger1000 communication address is 247.

When the logger forwards data from a device (e.g. inverter), the device (inverter) address is its forwarding address. For information on how to get the address and access data, please refer to "Logger Modbus Protocol Data Forwarding Guide".

II. Physical Interface

1. RS485

	Default setting
Addressing	247
Broadcast	Yes
Baud rate	9600 bit/s
Check	N/A
Data bit	8
Stop bit	1
Mode	RTU
Electrical interface	RS485-2W cable connection

2. Ethernet (for the Logger1000, default IP: 12.12.12.12, default open port: 502; for the Logger3000, default ETH1 IP: 12.12.12.12, default ETH2 IP: 13.13.13.13, default open port: 502)

III. Communication Description

1. Data type

U16 - 16-bit unsigned integer, big-endian.

S16 - 16-bit signed integer, big-endian.

U32 - 32-bit unsigned integer; little-endian for double-word data; big-endian for byte data.

S32 - 32-bit signed integer; little-endian for double-word data; big-endian for byte data.

U64 - unsigned 64-bit integer data, big-endian format.

S64 - signed 64-bit integer data, big-endian format.

For example,

U16 data 0x0102 is transmitted in the order 01, 02.

U32 data 0x01020304 is transmitted in the order 03, 04, 01, 02.

U64 data 0x0102030405060708 is transmitted in the order 01, 02, 03, 04, 05, 06, 07, 08.

The address register starts counting at 1, so the communication address = protocol address - 1.

2. Numerical description

Decimal numbers are transmitted as integers after expansion. For example,

10.333KW is transmitted as 10 333 in expanded form; 800.5V is transmitted as 8005.

Negative numbers are transmitted as complements. For example, 0xFFFF denotes -1.

"Reserved" or unsupported registers cannot be queried or set. "F"s are returned when unsigned numbers are queried. For example, "0xFFFF" is returned for U16,

"0xFFFFFFFF" for U32, and "0xFFFFFFFFFFFFFFFF" for U64. The largest positive

numbers are returned for signed numbers, for example, "0x7FFF" is returned for S16,

"0x7FFFFFFF" for S32, "0x7FFFFFFFFFFFFFFF" for S64, and 0x00 for UTF-8.

UTF-8 occupies 1 byte. An odd length is padded with 0x00.

3. Address type

The input register is a read-only register and supports 0x04 function code read.

The holding register is a read/write register and supports 0x03 function code query

and 0x06/0x10 function code write.

4. Check type

CRC16, polynomial 0xA001, little-endian.

IV. Address Definition Table

4.1 Input register (address type: 3X)

Note: In the Remarks column, "Logger1000" indicates that it is supported by the Logger1000 only; "Logger3000" indicates that it is supported by the Logger3000 only; blank indicates that it is supported by both Logger1000 and Logger3000.

Name	Address	Data type	Factor	Unit	Remarks
Device type code	8000	U16	1		0x0705 Logger3000 0x0710 Logger1000
Protocol number	8001	U32	1		
Communication protocol version	8003	U32	1		
Total devices connected	8005	U16	1	Set	
Total faulty devices	8006	U16	1	Set	
Reserved	8015	U32	1		
Reserved	8017	U16	1		
Digital input state	8021	U32	1		Currently only the low 16 bits are used. Logger1000 only uses 8 bits, while Logger3000 uses 16 bits
Reserved	8023	U32	1		
Reserved	8025	U32	1		
PT100-1	8027	S16	0.1	□	Logger3000
PT100-2	8028	S16	0.1	□	Logger3000
ADC1 voltage	8029	S16	0.01	V	
ADC1 current	8030	S16	0.01	mA	
ADC2 voltage	8031	S16	0.01	V	
ADC2 current	8032	S16	0.01	mA	
ADC3 voltage	8033	S16			Shared by Logger3000 and Logger1000, where Logger3000 consumes 0.01mV, while Logger1000 consumes 0.01V
ADC4 voltage	8034	S16			Shared by Logger3000 and Logger1000, where Logger 3000 consumes 0.01mV while Logger 1000 consumes 0.01V
ADC3 current	8035	S16	0.01	mA	Logger1000
ADC4 current	8036	S16	0.01	mA	Logger1000
Reserved	8037	S16	1		
Longitude	8054	S32	0.0001	°	Logger3000
Latitude	8056	S32	0.0001	°	Logger3000
Max. total nominal active power	8058	U16	1	kW	

Min. total nominal active power	8059	U16	1	kW	
Max. total nominal reactive power	8060	U16	1	kvar	
Min. total nominal reactive power	8061	S16	1	kvar	
Inverter preset total active power	8066	U16	1	kW	
Inverter preset total reactive power	8067	S16	1	kvar	
Logger On/Off state	8068	U16	1		0: Off 1: On
Logger unlatch state	8069	U16	1		0: latched 1: unlatched
Total active power	8070	U64	1	W	
Daily power yield	8074	U32	0.1	kWh	
Total reactive power	8076	S64	1	var	
Total power yield	8080	U64	0.1	kWh	
Min. adjustable active power	8084	U32	0.1	kW	
Max. adjustable active power	8086	U32	0.1	kW	
Min. adjustable reactive power	8088	S32	0.1	kvar	
Max. adjustable reactive power	8090	S32	0.1	kvar	
Nominal active power	8092	U32	0.1	kvar	
Nominal reactive power	8094	U32	0.1	kvar	
Grid-connected devices	8096	U16	1	Set	
Off-grid devices	8097	U16	1	Set	
Monthly yield of array	8098	U64	0.1	kWh	
Annual yield of array	8102	U64	0.1	kWh	
Apparent power of array	8106	U64	1	VA	

4.2 Holding register (address type: 4X)

Note: The holding register is set to support single function only. All commands from the broadcast address 0 are directly transparently transmitted to the inverter.

Name	Address	Data type	Upper limit	Lower limit	Factor	Unit	Remarks
Set On/Off for subarray inverter	8002	U16					0: Off 1: On
Set the active power value for subarray inverter	8003	U32			0.1	kW	
Set the active power ratio for subarray inverter	8005	U32			0.1	%	
Set the reactive power value for subarray inverter	8007	S32			0.1	kvar	
Set the reactive power ratio for subarray inverter	8009	S32			0.1	%	
Set the power factor	8011	S32			0.001		

for subarray inverter							
D01	8013	U16			1		Logger3000
D02	8014	U16			1		0: Common port and
D03	8015	U16			1		NC port are closed
D04	8016	U16			1		1: Common port and NO port are closed

V. Examples

6.1. The background communicates with the logger over the network. The address of the logger is 247. When the background is bound to the port 502 of the logger, it needs to query the DI state of the logger.

Background sends: 09 79 00 00 00 06 F7 04 1F 54 00 02

Logger replies: 09 79 00 00 00 07 F7 04 04 00 01 00 00

In the data replied, "09 79 00 00 00 06" and "09 79 00 00 00 07" are headers of the Modbus TCP message; F7 is the address of the logger; 04 is the function code; 0x1F54 = 8020 is to query the data of Register 8021. According to this protocol, Register 8021 indicates the "digital input state (DIN)", and the data of Register 8021 in the logger reply message is 0x0000.

6.2. Set boot

Send command: 0A B3 00 00 00 06 F7 06 1F 41 00 01

Reply data: 0A B3 00 00 00 06 F7 06 1F 41 00 01

In the command sent, "0ab3000006" is the Modbus TCP message header, F7 is the logger address, 06 is the function code, 0x1F41 = 8001 is to write data to Register 8002; the value to write is 0x0001 indicates booting.

6.3 Set the active power ratio for subarray inverter

Send command: 00 00 00 00 00 0b F7 10 1F 42 00 02 04 00 00 00 00

Reply data: 00 00 00 00 00 06 F7 10 1F 42 00 02

In the command sent, "00 00 00 00 00 0b" is the Modbus TCP message header, F7 is the Logger address, 10 is the function code, 0x1F42 = 8002 is to write two consecutive register data to the starting address of Register 8003, and the value to write is 0x00000000.

6.4. Set the active power ratio for subarray inverter

Send command: 00 00 00 00 00 0b F7 10 1F 44 00 02 04 01 5E 00 00

Reply data: 00 00 00 00 00 06 F7 10 1F 44 00 02

In the command sent, "00 00 00 00 00 0b" is the Modbus TCP message header, F7 is the Logger address, 10 is the function code, 0x1F44 = 8004 is to write two consecutive register data to the starting address of Register 8005, and the value to write is

PŘÍLOHA P II: PYTHON SCRIPT

 main.py

```
1 import requests
2 from bs4 import BeautifulSoup
3 from pymodbus.client.sync import ModbusTcpClient
4 from datetime import datetime
5 from time import sleep
6
7 DATE_CZ = datetime.now().strftime("%Y-%m-%d")
8 HOUR_NOW = datetime.now().strftime("%H")
9 SERVER_IP_ADDRESS = "xxx.xxx.xxx.xxx"
10 SERVER_PORT = 503
11
12 def get_data_CZ(date_cz):
13     attempts = 0
14     url = f"https://www.ote-cr.cz/cs/kratkodobe-trhy/elektrina/denni-trh?date={date_cz}"
15     while attempts < 5:
16         try:
17             session = requests.get(url)
18             if session.status_code == 200:
19                 html = BeautifulSoup(session.text, "html.parser")
20                 tables = html.find_all("tbody")[1]
21                 columns = tables.find_all("tr")
22                 values = {str(int(column.find("th").text) - 1).zfill(2):
23                          int(column.find("td").text.strip().replace(",","")) for column in columns[:24]}
24                 if len(values) == 24:
25                     return values
26             except Exception as e:
27                 print(f"Error during the data fetching process: {e}")
28                 attempts += 1
29                 print(f"Error downloading data from the website, attempt #{attempts}.")
30                 sleep(2)
31     return None
32
33 def write_data(server_ip, port):
34     client = ModbusTcpClient(host=server_ip, port=port)
35     if client.connect():
36         try:
37             data = get_data_CZ(DATE_CZ)
38             if data is None:
39                 print("Failed to download data from the website.")
40                 return
41             client.write_register(1, data[ HOUR_NOW ], unit=1)
42         except Exception as e:
43             print(f"Error writing to Modbus: {e}")
44         finally:
45             client.close()
46     else:
47         print("Failed to connect to the Modbus.")
48
49 write_data(SERVER_IP_ADDRESS, SERVER_PORT)
50
```


PŘÍLOHA P III: MODBUS REGISTRY SOLAREEDGE



Address	Size	Name	Type	Description
40001	2	C_SunSpec_ID	uint32	Value = "SunS" (0x53756e53). Uniquely identifies this as a SunSpec MODBUS Map
40003	1	C_SunSpec_DID	uint16	Value = 0x0001. Uniquely identifies this as a SunSpec Common Model Block
40004	1	C_SunSpec_Length	uint16	65 = Length of block in 16-bit registers
40005	16	C_Manufacturer	String(32)	Value Registered with SunSpec = "SolarEdge "
40021	16	C_Model	String(32)	SolarEdge Specific Value
40045	8	C_Version	String(16)	SolarEdge Specific Value
40053	16	C_SerialNumber	String(32)	SolarEdge Unique Value
40069	1	C_DeviceAddress	uint16	MODBUS Unit ID

Inverter Device Status Values

The following **I_Status_XXXX** values are supported:

Parameter	Value	Description
I_STATUS_OFF	1	Off
I_STATUS_SLEEPING	2	Sleeping (auto-shutdown) – Night mode
I_STATUS_STARTING	3	Grid Monitoring/wake-up
I_STATUS_MPPT	4	Inverter is ON and producing power
I_STATUS_THROTTLED	5	Production (curtailed)
I_STATUS_SHUTTING_DOWN	6	Shutting down
I_STATUS_FAULT	7	Fault
I_STATUS_STANDBY	8	Maintenance/setup

Inverter Model MODBUS Register Mappings

The following table lists the supported MODBUS register values.

Unsupported values are indicated by the NOT_IMPLEMENTED value.

DC values are not supported for three phase inverters with synergy technology.

The base register of the Device Specific block is set to 40070 (MODBUS PLC address [base 1]), or 40069 (MODBUS Protocol Address [base 0]).

- **acc32** is uint32 that should always increase. Its value is in the Range of 0 ...4294967295.

■ Scale Factors

As an alternative to floating point format, values are represented by Integer values with a signed scale factor applied. The scale factor explicitly shifts the decimal point to left (negative value) or to the right (positive value).

For example, a value "Value" may have an associated value "Value_SF"

Value = "Value" * 10^{Value_SF} for example:

For "Value" = 2071 and "Value_SF" = -2 Value = 2071*10⁻² = 20.71

For "Value" = 2071 and "Value_SF" = 2 Value = 2071*10² = 207100

Address	Size	Name	Type	Units	Description
40070	1	C_SunSpec_DID	uint16		101 = single phase 102 = split phase ¹ 103 = three phase
40071	1	C_SunSpec_Length	uint16	Registers	50 = Length of model block
40072	1	I_AC_Current	uint16	Amps	AC Total Current value
40073	1	I_AC_CurrentA	uint16	Amps	AC Phase A Current value

Address	Size	Name	Type	Units	Description
40074	1	I_AC_CurrentB	uint16	Amps	AC Phase B Current value
40075	1	I_AC_CurrentC	uint16	Amps	AC Phase C Current value
40076	1	I_AC_Current_SF	int16		AC Current scale factor
40077	1	I_AC_VoltageAB	uint16	Volts	AC Voltage Phase AB value
40078	1	I_AC_VoltageBC	uint16	Volts	AC Voltage Phase BC value
40079	1	I_AC_VoltageCA	uint16	Volts	AC Voltage Phase CA value
40080	1	I_AC_VoltageAN ²	uint16	Volts	AC Voltage Phase A to N value
40081	1	I_AC_VoltageBN ¹	uint16	Volts	AC Voltage Phase B to N value
40082	1	I_AC_VoltageCN ¹	uint16	Volts	AC Voltage Phase C to N value
40083	1	I_AC_Voltage_SF	int16		AC Voltage scale factor
40084	1	I_AC_Power	int16	Watts	AC Power value
40085	1	I_AC_Power_SF	int16		AC Power scale factor
40086	1	I_AC_Frequency	uint16	Hertz	AC Frequency value
40087	1	I_AC_Frequency_SF	int16		Scale factor
40088	1	I_AC_VA	int16	VA	Apparent Power
40089	1	I_AC_VA_SF	int16		Scale factor
40090	1	I_AC_VAR	int16	VAR	Reactive Power
40091	1	I_AC_VAR_SF	int16		Scale factor
40092	1	I_AC_PF	int16	%	Power Factor
40093	1	I_AC_PF_SF	int16		Scale factor
40094	2	I_AC_Energy_WH	acc32	WattHours	AC Lifetime Energy production
40096	1	I_AC_Energy_WH_SF	uint16		Scale factor
40097	1	I_DC_Current	uint16	Amps	DC Current value
40098	1	I_DC_Current_SF	int16		Scale factor
40099	1	I_DC_Voltage	uint16	Volts	DC Voltage value
40100	1	I_DC_Voltage_SF	int16		Scale factor
40101	1	I_DC_Power	int16	Watts	DC Power value
40102	1	I_DC_Power_SF	int16		Scale factor
40104	1	I_Temp_Sink	int16	Degrees C	Heat Sink Temperature
40107	1	I_Temp_SF	int16		Scale factor
40108	1	I_Status	uint16		Operating State
40109	1	I_Status_Vendor	uint16		Vendor-defined operating state and error codes. For error description, meaning and troubleshooting, refer to the SolarEdge Installation Guide.

Meter Models

The SunSpec Alliance Interoperability Specification describes the data models and MODBUS register mappings for meter devices used in Renewable Energy systems. This section defines the models for:

- Single Phase Meter
- Split Phase Meter
- WYE (4-wire) Meter
- Delta (3-wire) Meter

² Supported only in split-phase configurations (Japanese grid and 240V grid in North America).

PŘÍLOHA P IV: MODBUS REGISTRY ALPHAESS



Smarten Your Energy

3. Parameter address table

Address Register	variable	Belong to R/W	Data format	Data Model	Remarks
System Running Data					
08D4H 08D5H	System fault	RO	Occupy 4 byte	unsigned int	<u>Note1</u>
Industry Remote Control Parameter					
4002H 4003H	AC power setting	R/W	Occupy 4 byte	int	1W/Bit
4004H~ 4008H	Reserved				
4009H	Mode on/off	R/W	Occupy 2 byte	unsigned short	0: Mode off 1: Mode on
400AH	Reserved				
400BH	Clear fault	R/W	Occupy 2 byte	unsigned short	0: False 1: True
Industry Local Control Parameter					
40B3H	Local remote mode	R/W	Occupy 2 byte	unsigned short	0: Local mode 1: Remote mode
Industry Meter					
4D00H	CT Enable(Grid meter)	RO	Occupy 2byte	unsigned short	1/bit
4D01H	CT Rate(Grid meter)	RO	Occupy 2byte	unsigned short	1/bit
4D02H	PT Enable(Grid meter)	RO	Occupy 2byte	unsigned short	1/bit
4D03H	PT Rate(Grid meter)	RO	Occupy 2byte	unsigned short	1/bit
4D04H 4D05H	Total energy feed to grid(Grid meter)	RO	Occupy 4 byte	unsigned int	0.01kWh/bit
4D06H 4D07H	Total energy consume from grid(Grid meter)	RO	Occupy 4 byte	unsigned int	0.01kWh/bit
4D08H	Voltage of A phase(Grid meter)	RO	Occupy 2 byte	unsigned short	1V/bit
4D09H	Voltage of B phase(Grid meter)	RO	Occupy 2 byte	unsigned short	1V/bit
4D0AH	Voltage of C phase(Grid meter)	RO	Occupy 2 byte	unsigned short	1V/bit
4D0BH	Current of A phase(Grid meter)	RO	Occupy 2 byte	short	0.1A/bit
4D0CH	Current of B phase(Grid meter)	RO	Occupy 2 byte	short	0.1A/bit

			2 byte		
4D0DH	Current of C phase(Grid meter)	RO	Occupy 2 byte	short	0.1A/bit
4D0EH	Frequent(Grid meter)	RO	Occupy 2 byte	unsigned short	0.01Hz
4D0FH 4D10H	Active power of A phase(Grid meter)	RO	Occupy 4 byte	int	1W/bit
4D11H 4D12H	Active power of B phase(Grid meter)	RO	Occupy 4 byte	int	1W/bit
4D13H 4D14H	Active power of C phase(Grid meter)	RO	Occupy 4 byte	int	1W/bit
4D15H 4D16H	Total Active power(Grid Meter)	RO	Occupy 4 byte	int	1W/bit
4D17H 4D18H	Reactive power of A phase(Grid meter)	RO	Occupy 4 byte	int	1var/bit
4D19H 4D1AH	Reactive power of B phase(Grid meter)	RO	Occupy 4 byte	int	1var/bit
4D1BH 4D1CH	Reactive power of C phase(Grid meter)	RO	Occupy 4 byte	int	1var/bit
4D1DH 4D1EH	Total reactive power(Grid meter)	RO	Occupy 4 byte	int	1var/bit
4D1FH~ 4D2AH	Reserved		Occupy 24 byte		
4D2BH~ 4D7FH	Reserved(Grid meter)		Occupy 170 byte		
4D80H	CT Enable(Pv meter)	RO	Occupy 2byte	unsigned short	1/bit
4D81H	CT Rate(Pv meter)	RO	Occupy 2byte	unsigned short	1/bit
4D82H	PT Enable(Pv meter)	RO	Occupy 2byte	unsigned short	1/bit
4D83H	PT Rate(Pv meter)	RO	Occupy 2byte	unsigned short	1/bit
4D84H 4D85H	Total energy feed to grid(Pv meter)	RO	Occupy 4 byte	unsigned int	0.01kWh/bit
4D86H 4D87H	Total energy consume from grid(Pv meter)	RO	Occupy 4 byte	unsigned int	0.01kWh/bit
4D88H	Voltage of A phase(Pv meter)	RO	Occupy 2 byte	unsigned short	1V/bit
4D89H	Voltage of B phase(Pv meter)	RO	Occupy 2 byte	unsigned short	1V/bit
4D8AH	Voltage of C phase(Pv meter)	RO	Occupy 2 byte	unsigned short	1V/bit