VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2019

Bc. Martin Váňa

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

KYBERNETICKÉ PROSTŘEDÍ PRO SYSTÉMY TYPU ICS/SCADA
CYBER-ENVIRONMENT FOR SYSTEMS OF ICS/SCADA TYPE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Martin Váňa**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Radek Fujdiak, Ph.D.**
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Martin Váňa ***ID:*** 173774
Ročník: 2 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Kybernetické prostředí pro systémy typu ICS/SCADA

POKYNY PRO VYPRACOVÁNÍ:

Student bude mít za úkol zanalyzovat komunikační model ICS/SCADA, následně vytvořit přehled komunikačních protokolů s úzkým zaměřením na protokoly MODBUS, DNP 3, IEC 60870-5-104 a IEC 61850 (GOOSE, MMS a SMV). Výsledkem analýzy protokolů by měl být také přehled dostupných implementací a řešení využitelných pro testovací prostředí a seznámení se s již hotovou implementací v laboratorním přípravku pro IEC61850. Analýza bude dále obsahovat i přehled dostupných nástrojů na vizualizaci dat (otevřené řešení jako Free Scada, IndigoSCADA, openDAX, openSCADA, S.E.E.R. 2, SCADA Process Viewer, ScadaBR, Szarp či komerční jako PROMOTIC a další – doporučen openSCADA), který vyústí ve výběr jedné vhodné implementace. Následně proběhne návrh a realizace testovacího kybernetického prostředí, kdy nízkoúrovňové prvky budou uvažovány jako simulované/virtualizované. Vytvořené prostředí by mělo být uživatelsky jednoduché, modulární a lehce modifikovatelné (rozšířitelné).

DOPORUČENÁ LITERATURA:

[1] „SCADA Systems for Industry“. IDC Technologies. Revize 3.1, Praktikum. 2010.

[2] ČSN EN 61850. „Komunikační sítě a systémy v podřízených stanicích“. Úřad pro technickou normalizaci, metrologii a státní zkušebnictví.

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 16.5.2019

Vedoucí práce: Ing. Radek Fujdiak, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato práce se věnuje problematice kybernetického prostředí pro systémy typu ICS/SCADA. První, kratší část je zaměřena na obecné představení ICS/SCADA systémů a jejich fungování. Je zde rozebrán jejich komunikační model a elementární prvky nezbytné pro fungování celého systému. Tato část je převážně teoretická a slouží jako úvod do problematiky. Je nezbytná pro pochopení druhé, praktické části. V rámci praktické části je jako první vybírán vhodný systém pro realizaci celého projektu. Jsou zde definována kritéria na základě, kterých je realizován samotný systém. Systém je zprovozněn pod „frameworkem“ openMUC a testován pomocí simulátorů na základě zadání této práce.

KLÍČOVÁ SLOVA

openMUC, SCADA systémy, ICS, RTU, PLC, MTU, IEC 61850, ModBus, IEC 60870, Kybernetické prostředí.

ABSTRACT

The thesis explores the problematics of cyber environment for the ICS/SCADA systems. First, shorter section is mainly focused on general introduction into the ICS/SCADA systems and their inner workings. Communication model of a general SCADA system and its foundational elements are explained. It is mainly theoretical passage and it serves as an introduction. It is necessary for understanding the second part which is mainly practical. The appropriate system is chosen as a first thing in the practical part of the thesis for the implementation of the whole project. There are defined criteria on which the system itself is implemented. Following that the system itself is implemented under a framework called openMUC and it is tested with help of the simulators according to the objective of the thesis.

KEYWORDS

openMUC, SCADA systems, ICS, RTU, PLC, MTU, IEC 61850, ModBus, IEC 60870, Cyber-Environment.

VÁŇA, Martin. *Kybernetické prostředí pro systémy typu ICS/SCADA*. Brno, 2019, 60 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Radek Fujdiak, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Kybernetické prostředí pro systémy typu ICS/SCADA“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Radku Fujdiakovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno ....

podpis autora

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj
pro inovace

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Průmysl 4.0 se stává čím dál více skloňovaným pojmem. Teoreticky by se mělo jednat o další revoluci ve výrobě, zpracování a skoro by se dalo říci v přemýšlení. Přeci jenom, když přišla tzv. 3. průmyslová revoluce, která je definovaná vynálezem PLC – *Programmable logic controller – Programovatelný logický automat* (PLC)[1], tak si také nikdo nedokázal představit, co všechno přinese. Dnes automatizace vládne průmyslu. V rámci 4. revoluce se hlavně jedná o rozsáhlou digitalizaci zavedených systémů. Digitalizace umožňuje automatizaci na úplně nové úrovni. Ale ani automatizované systémy nemohou zůstat bez dozoru a řízení v případě potřeby. Právě zde se vytváří místo pro systémy typu ICS *Industrial control system – Industriální systémy řízení* (ICS)/SCADA *Supervisory control and data acquisition – Dispečerské řízení a sběr dat* (SCADA)[3].

ICS je spíše obecný pojem, který představuje rodinu různých implementací řízení. Může se jednat o velmi malé systémy o 20 servomotorcích až po značně rozsáhlejší systémy obsahující čerpadla, čidla, reservoáry vody apod. například v chladicím okruhu jaderné elektrárny. Přesně pod tyto systémy řízení spadá i SCADA[2]. SCADA je již konkrétnější pojem. Představuje architekturu, „šablonu“ řídícího systému. Architektura počítá s využitím klasických počítačů od PC po servery, síťového propojení jednotlivých prvků infrastruktury SCADA a rozhraní člověk-stroj, jinak také *Human-machine interface – Rozhraní člověk-stroj* (HMI)[3].

Všechny části tohoto komplikovaného sytému budou postupně podrobně přiblíženy. V první části práce je věnován prostor elementárním prvkům systému SCADA, jejich využití, architektuře a základnímu vysvětlení jejich fungování. Například je prostor věnován PLC automatům nebo celé komunikační infrastruktuře (na čem je postavena atd.). Ve druhé části práce jsou přiblíženy protokoly užívané ve SCADA systémech. Základní vysvětlení jejich fungování, jejich specifika a využití ve jednotlivých částech celého systému. Hlavní pozornost je věnována 4 základním protokolům nejvíce využívaných v dnešních systémech i systémech budoucích. Primárně jde o MODBUS, DNP 3, IEC 60870-5-104, IEC 61850[3].

Další, již praktická část se věnuje nástrojům pro vizualizaci dat. Jsou zde srovnány různé systémy od otevřených po uzavřená a komercializovaná řešení. Například jsou zde uvedeny systémy jako OpenSCADA, mySCADA, openMUC atd. Posléze je věnován prostor metodice zpracování praktického řešení, na kterou plynule přechází realizace kybernetického prostředí pod jedním z diskutovaných systémů. V rámci této realizace je rozebráno podrobněji fungování systému a jsou implementovány potřebné protokoly k funkčnosti celého systému, které jsou poté testovány.

1 Komunikační model ICS/SCADA

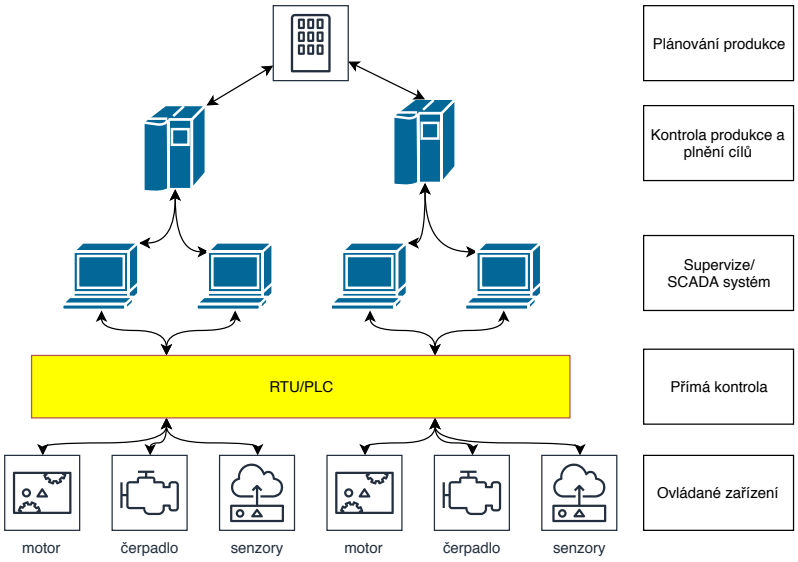
Na začátek práce si je potřeba nejdříve definovat, co představuje SCADA. Jak již bylo zmíněno v úvodu práce SCADA je součástí rodiny ICS systémů. SCADA sytém samotný představuje kombinaci získávání dat a tvoření datových struktur ze získaných dat. Plus na základě těchto dat, je možné skrze SCADA systém řídit celý ekosystém[4].

V praxi to znamená, že na koncové jednotky jsou napojeny kontrolní stanice sbírající data. Data jsou odeslány do řídícího centra, kde je v rámci systému provedena analýza a automatická kontrola i řízení. Na základě výstupů z analýzy jsou prezentovány sbíraná data a automatizovaně prováděné úkoly do hlavních řídících obrazovek, jenž jsou kontrolovány lidským operátorem. Pokud je potřeba zásah do systému lidským operátorem, případně jsou prováděny automatizované příkazy, tak jsou tyto informace šířeny zpět ke kontrolním stanicím a dále ke koncovým prvkům[5].

Jedná se tedy o hierarchický systém, který obsahuje mnoho prvků, takže je nutná jistá decentralizace a autonomie jednotlivých vrstev. Jednotlivé decentralizované clustery se ale sbíhají do hlavní centrály, kde se tvoří ucelený obrázek o stavu a řízení systému. Na základě zmíněného se tedy dají definovat jisté úrovně řízení. Úrovni je celkem 5[5]. Jde o nejhrubší rozdělení celé SCADA infrastruktury. Dělení je tedy zhruba takto:

- Plánování (Production scheduling),
- Kontrola produkce (Production control),
- Supervize (Plant supervisory),
- Přímá kontrola (Direct control),
- Ovládané zařízení (Field level).

Pro přehlednost je níže obrázek ilustrující těchto 5 úrovní řízení. První dvě úrovně (Plánování a kontrola produkce) nespadají přímo do kontroly procesů a dalo by se je zařadit spíše jako pomyslný management. Na těchto úrovní se tudíž neděje nic přímo relevantního s požadavky na tuto práci. Jsou zde pouze uvedeny pro úplnost. 3. úroveň je již naopak nezbytnou součástí kontroly procesů přímo na místě. Obsahuje instalaci SCADA systému a infrastrukturu, na které systém operuje. Je zde počátek infrastruktury celého systému. Dá se říci, že se zde nachází mozek celého řízení. O úroveň níže se již nacházejí kontrolní jednotky, které tvoří jádro. Slouží jako nervový systém pro 3. úroveň. Poslední úroveň představuje řízené jednotky a senzory. Celý systém je propojený síťovou infrastrukturou. Může se jednat například o TCP/IP architekturu[5].



Obr. 1.1: Komunikační model SCADA[6].

1.1 Elementární prvky

Výše je zmíněn pouze obecný popis systému. Pro pochopení fungování celého systému do nejmenšího detailu je potřeba si definovat elementární součásti. Každý systém se skládá z různých prvků a je tak svým způsobem unikátní. V rámci komunikačního modelu SCADA systému by ale se dalo definovat 5 hlavních součástí, které se nachází v systému „povinně“[5]:

- Supervisory computers,
- RTU – Remote terminal units,
- PLC – Programmable logic controller,
- Communication infrastructure,
- HMI – Human-machine interface.

1.1.1 Supervisory computers

Počítače, které lze nazvat srdcem celého systému. Doslovný překlad by mohl být „Dohledové počítače“. Zde se všechny data posílají a zpracovávají. Zároveň jsou odsud vysílány všechny příkazy k potřebným prvkům topologie. U těchto počítačů se nachází lidský operátor. Pozoruje, případně řídí všechny procesy spadající pod celý ekosystém. Právě na tomto prvku topologie běží potřebný software k řízení pomyslné továrny. K tomu aby mohla být továrna odsud řízena, se sem sbíhají všechny spoje a předávají informace. Tudíž pravě odtud je možné řídit všechny PLC a RTU jednotky[5]. Vše přehledně centralizované v jednom HUBu, jaký lze například vidět viz obr. 1.2. Pokud se jedná o menší SCADA systémy, tak může být Dohledový

Obr. 1.2: Řídící centrum Kuvajtské elektrárny[7].

počítač pouze jeden. V tomto počítači jsou agregovány všechny potřebné softwary a funkce pro hladký běh celého systému. V případě, že se jedná o velký SCADA systém, tak jsou většinou role rozděleny do různých počítačů. Například servery obstarávající získávání dat mohou být odděleny od počítače, kde operuje grafické rozhraní SCADA. Těchto počítačů může být několik, jak je vidět na obrázku výše.

Nebo například severy obstarávající obnovení funkcí po nehodě mohou být distribuované na několika místech, právě pro zvýšení bezpečnosti. Mohou zde být také zapojeny redundantní servery, které se zapojí okamžitě, jakmile právě fungující servery selžou. Redundantní servery okamžitě (v rámci milisekund až sekund) nahradí funkci selhávajícího serveru. Jedná se o naprosto nezbytnou funkci v časově kritických aplikacích (například nukleární elektrárny), kde nesmí dojít k výpadku[5]. To vše obsahuje pojem Supervisory computer.

Dohledové počítače mají dvě podkategorie[2]:

- **Control server – Kontrolní server,**
  Na tomto severu se nachází software, který komunikuje s PLC prvky v síti. Neboli software, který je designovaný na **získávání** dat ze zařízení blíže k 5. úrovni SCADA systému (viz výše).
- **SCADA server/Master terminal unit.**
  Tento systém se chová jako hlavní systém tzv. Master ve SCADA prostředí. RTU a PLC jednotky jsou **ovládané** odtud a chovají se jako tzv. slave.

1.1.2 Remote terminal units

Remote terminal unit nebo také Remote telemetry unit je speciální prvek sítě zaměřený na získávání dat a ovládaní prvků na něho napojených. Je navržený, tak aby podporoval SCADA systémy. Jedná se nízkoúrovňový prvek, téměř koncový prvek. Na něj jsou napojeny již pouze koncové čidla a jsou přítomny přímo na místě vybaveny nějakým způsobem komunikace. Může jít přímo o fyzické napojení vodičem nebo bezdrátově. Pokud jde o bezdrátovou variantu, tak musí RTU obsahovat anténu a vše potřebné pro její běh. Tato varianta se využívá primárně u nedostupných míst vzhledem k nevýhodám, které bezdrátový provoz přináší. Může se jednat o problémy se spolehlivostí provozu. Zařízení je z podstaty komplikovanější. Je zde možnost různých interferencí. Je potřeba také mít zdroj energie, což přináší další komplikace a podobně[8].

Obr. 1.3: RTU jednotky od SCADAservis.cz[9].

Jednotky mají využití v různých aplikacích průmyslu. Jde například o řízení odpadních systémů nebo ropovody, případně řízení výroby energie[9]. Možnosti jsou široké. S tím souvisí specifikum, že každé zařízení je specifické a navržené na určitou specifickou funkci.

Moderní RTU jednotky obsahují jak analogové, tak digitální input/output rozhraní. Mohou používat obě rozhraní pro komunikaci s koncovými čidly či naopak s Master jednotkou. S čímž také souvisí, že RTU jednotky mohou být napojeny na více Master jednotek[8]. Což může být užitečné při výpadku nebo při potřebě informací pro více zdrojů.

Jednotky jsou schopny komunikovat přes Ethernet nebo přes Serial port. Záleží na implementaci. Jsou schopné přijímat příkazy, ale zároveň mají jistou autonomii. Je zde tudíž možnost, že jednotka vykonává základní funkce autonomně. Rozšířené možnosti autonomie pak již zahrnují PLC, kterým je věnována následující část.

1.1.3 Programmable logic controller

PLC by se dal nejlépe označit za malý průmyslový počítač, který byl původně zamýšlen pro provádění logických funkcí za pomocí elektricky spínaných obvodů, jako jsou například relé nebo mechanické časovače atd. Od té doby ale prodělaly vývoj a to dost značný. Dnes jsou PLC rozvinuté kontroléry schopné ovládat komplexní procesy na 4. úrovni SCADA systému. Z čehož vyplývá, že jsou značně využívány právě ve SCADA systémech pro distribuci informací a zároveň ovládání větších celků[10].

Obr. 1.4: Siemens PLC modul[11].

Pokud srovnáme obrázky 1.3 a 1.4 je na první pohled patrné, že PLC jsou mnohem vyspělejší zařízení. Mohou obstarávat mnohem více funkcí. Paradoxně k předchozímu jsou to právě PLC, které jsou často užívány na místo RTU. RTU jednotka je velmi úzce zaměřena na určitou funkci. Naproti tomu PLC jsou flexibilní a je možné je překonfigurovat na mnohé funkce, kterých by RTU nikdy nedosáhla, protože na to nebyla navržena[10]. Z toho důvodu zde lze mluvit i o jisté ekonomičnosti využití PLC místo RTU, přestože počáteční investice je vyšší. Pokud je ztracena potřeba pro vykonávání funkce RTU, například část produkční linky je zrušena, tak RTU může být vyhozena, protože není možné ji použít nikde jinde (nebo nepraktické).

PLC stačí přeprogramovat na jinou potřebnou funkci a může být dále využíván[2]. Program může být psán v různých jazycích. Na začátku se jednalo primárně o BASIC a C[10]. Dnes jsou již vytvořeny speciální programy, pomocí kterých je mnohem jednodušší naprogramovat PLC modul. Využívají grafických rozhraní atd. Je zde ale nezbytnost, ke každému modulu od jiného výrobce přistupovat unikátně.

1.1.4 Communication infrastructure

Každý systém potřebuje komunikační infrastrukturu. Infrastrukturu, po které se budou informace přenášet. Této problematice se věnuje právě tato část. Každá z úrovní SCADA systému má svoje specifika a rozdílné síťové charakteristiky. Plus jsou zde také značné rozdíly závisející na implementaci vzhledem k tomu, že každý systém je prakticky unikátní a speciálně navržený na určitou aplikaci[14].

Obecně vzato by se dalo říci, že moderní implementace těchto systémů využívají velice podobné síťové struktury jako TCP/IP model[2]. Je zde snaha je, co nejvíce integrovat, aby do těchto systémů bylo, co nejjednodušší přistupovat. Je to z důvodu možnosti mobility operátora. Například inženýr se může připojit z jakéhokoliv místa a řešit nastalé problémy. Informace mohou být rychleji komunikovány. Nebo se management může připojit do sítě a okamžitě vyhodnocovat, zda vytyčené cíle budou dodrženy. Výčet možností je značný. Každá síť obsahuje vždy určité prvky jako[2]:

- **Fieldbus síť** Fieldbus zahrnuje síťové komunikační protokoly primárně zaměřené na průmyslovou aplikaci. Od 90. let jsou standardizované pod IEC 61158[12]. Hlavní funkcí této „sběrnice“ je propojovat senzory a podobná zařízení k PLC nebo jiným řídícím systémům. Značnou výhodou této technologie je obejití nezbytnosti propojení senzoru a jeho kontroléru v topologii Bod-Bod. Takže nemusí být veden vodič signálu ke každému senzoru zvlášť od kontroléru. Senzory s kontrolérem komunikují pomocí protokolu na např. potenciálně sdílené lince. Posílané zprávy pak mají informace o zdroji a cíli zprávy a podle toho jsou příkazy doručovány. Z toho lze vyvodit značnou simplifikaci celé sítě a omezení cenové a prostorové náročnosti.
- **Kontrolní síť** Tato síť se již nachází na úrovni propojení vyšších vrstev SCADA systému. Primárně propojuje supervizorní prvky sítě (např. počítač operátora) s PLC moduly, takže s kontrolními prvky, ale nižší úrovně.
- **Směrovač** Směrovače mají stejnou funkci, jako v každé síti architektury např. TCP/IP. Zde tudíž přenáší/směrují zprávy mezi sítěmi. Z hlediska SCADA systému mohou propojovat MTU a RTU jednotky dále do jádra sítě, aby byly lépe dostupné.
- **Modem** Modem je ve SCADA systému, ostatně i v jiných systémech, zařízení pro přeměnu mezi digitálním signálem a signálem vhodným pro přenos po standardní telefonní lince. Díky čemuž mohou zařízení mezi sebou komunikovat. Typické použití ve SCADA systémech jsou pro modem dálkové spoje mezi MTU a vzdálenými slave zařízeními. Pod jejich využití spadá zajištění vzdáleného přístupu pro řídící funkce, jako jsou například příkazy od operátora, měnění parametrů celé škály operací nebo diagnostika[2].

- **Firewall** Firewall slouží pro ochranu zařízení za ním. To samé dělá i v případě SCADA systému. Monitoruje a kontroluje jím procházející komunikaci, aby zamezil zneužití.
- **Vzdálené přístupové body** Vzdálené přístupové body jsou zařízení a místa Kontrolní sítě odkud lze vzdáleně nastavovat zařízení v topologii a získávat data. Může se jednat například o kontrolní tablety, notebooky, atd.

1.1.5 Human-machine interface

Ve své podstatě je Human-machine interface (Člověk-stroj rozhraní) software a hardware, který umožňuje operátorovi zobrazovat a monitorovat status běžících procesů pod SCADA systémem. Přes HMI se upravuje nastavení běžících procesů a je zde možnost zasahovat do automaticky ovládaného systému například v případě nouze[2]. HMI umožňuje nastavovat kontrolní algoritmy a parametry přímých kontrolérů. Z čehož plyne, že se odtud dá řídit celý systém do posledního detailu. Zobrazují se zde také informace o procesech, historie využití a procesů, souhrnné zprávy a ostatní podstatné informace. Co se považuje ze podstatné je upraveno přímo operátorem nebo designérem HMI. Přes toto rozhraní pak můžou přistupovat všichni autorizovaní uživatelé. Nemusí se jednat přímo o operátora, ale například i o administrátory, management nebo dokonce třetí strany s povolením[2].

Samozřejmě rozhraní se dá upravovat podle toho, kdo do systému přistupuje. Pokud se bude jednat o operátora, tak budou zobrazeny položky týkající se výroby, teploty zařízení nebo poruchy na pásu. V případě, že se připojí administrátor, tak bude mít vhled do různých nastavení systému, jako kdy se má sepnout čerpadlo, případně upustit ventil apod. A za předpokladu, že se připojí někdo z managementu, tak se mu budou zobrazovat spíše zprávy o produktivitě, výpadcích systému, efektivitě atd.

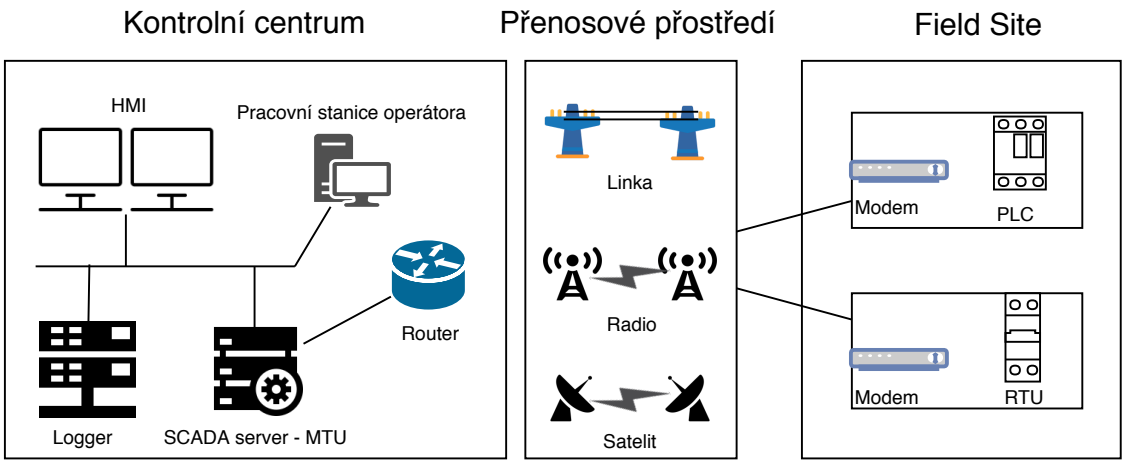
Místo, kde je systém využíván nebo platforma popřípadě rozhraní, jak už bylo výše zmíněno se může velmi lišit systém od systému. Prostředí pro laptop bude značně rozdílné než prostředí pro mobilní zařízení, které se bude také velmi lišit od prostředí využitém v kontrolním centru. Může se jednat také o webové prostředí. Možnosti jsou v této době značné a je pouze na návrháři celého systému, jak HMI pojme.

1.2 Základní uspořádání

Vzhledem k faktu, jak moc jsou jednotlivé systémy rozdílné, tak implementace každého SCADA ekosystému bude vypadat mírně rozdílně. Společné prvky zde ale nalézt lze. SCADA systémy zahrnují jak SW, tak HW. Existují zde tři globální oblasti, jak lze tento systém rozdělit do přehlednějšího systému. Na začátku máme Kontrolní středisko. Toto středisko obsahuje MTU, tedy přímo kontrolní server. Dále se zde nachází HMI a stanice operátorů. Pak je zde přítomný i logger tzv. Data historian. Tento prvek zajišťuje dlouhodobý sběr a uchování dat. To celé propojuje komunikační infrastruktura zakončená routery/směrovači vedoucího do přenosového média[13].

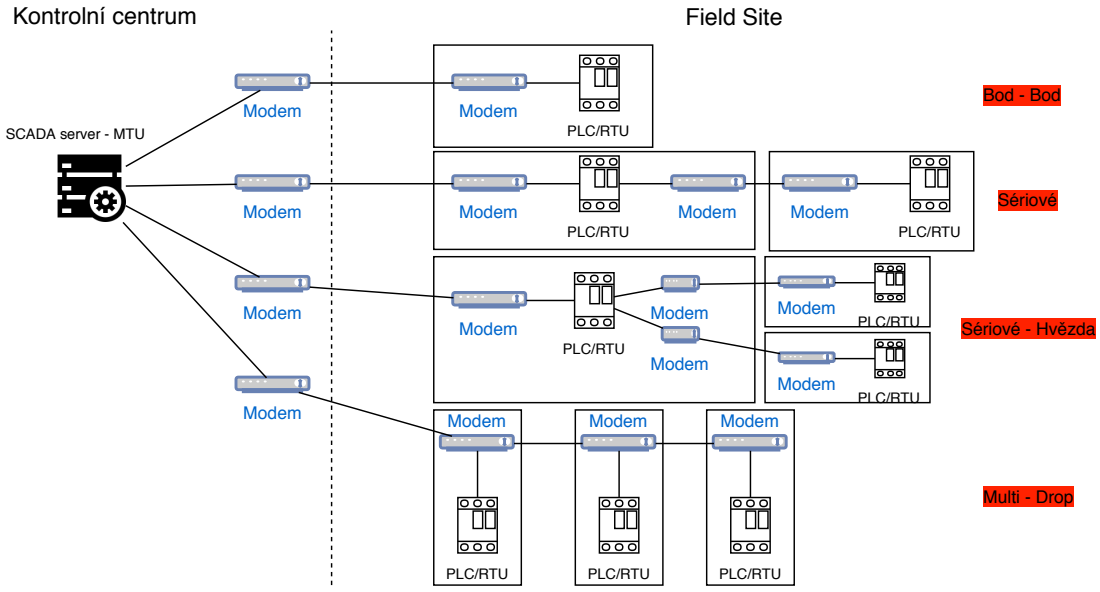
Ze směrovačů pak vychází další logický prvek SCADA topologie. Jedná se o přenosovou síť. Přenosová síť může využívat mnoho různých technologií. Například lze využít satelitní komunikaci, která ale bude mít vysoká zpoždění. Může se ukázat, jako nespolehlivá a náchylná na počasí, plus bude značně nákladná. Na druhou stranu její dosah je bezkonkurenční. Jako alternativu k této technologii je komunikace po pozemních vodičích neboli drátu. Tyto linky si lze celé pronajmout případně zainvestovat do vlastních, což sebou nese značnou finanční zátěž. Oproti satelitním systémům jsou podstatně spolehlivější a ve výsledku levnější. Jejich dosah je však velmi omezený. Kompromisem mezi zmíněnými předchozími technologiemi jsou mobilní sítě či jinak celulární architektura nebo obecně radiové spoje. Dalo by se říci, že ze zmíněných technologií mají střední dosah. Jsou omezeny pouze prostředím, kterým se síří, případně členitostí terénu. Fyzický spoj se musí stavět pouze k anténě, takže se šetří prostředky. Plus v dnešní době je tato technologie dobře zvládnutá, takže náročnost ustavení radiového spoje je minimální. Z čehož plyne menší náročnost na prostředky. Pokud jde o rychlé nasazení, tak je možné si pronajmout infrastrukturu od zaběhlých operátorů. Zmíněný výčet technologií je pouze orientační a pro jednoduchost velmi zkrácený. Pokud je potřeba více informací o propojování systémů, tak je lze nalézt v této knize[13].

Třetím a posledním logickým celkem jsou přímo řízená místa tzv. „field site“. Jedná se přímo o plochu továrny/elektrárny čehokoliv, na co je SCADA použitelná. Na pomyslném vstupu do tohoto logického celku se nachází modemy, které jak už bylo zmíněno, převádějí informace na kompatibilní signály mezi systémy. Těchto „field site“ je více druhů. Některé z nich jsou znázorněny na obrázku 1.5. Po modemu se již přímo nachází buď PLC nebo RTU jednotka. Záleží na systému a zvolenému přístupu. Zde se projevuje schopnost PLC jednotek emulovat funkce RTU jednotky, proto se „field site“ dělí na více možností[14].



Obr. 1.5: Obecná topologie SCADA systému[14].

Výše ukázaná obecná topologie má více iterací. V rámci komunikace SCADA server (MTU) a RTU (případně PLC) jsou využívaný podobné principy jako v TCP/IP modelu. Tedy existují různé implementace umístění a způsobu komunikace mezi těmito prvky. Jedná se například o architekturu Bod-Bod, sériové zapojení, sériové-hvězda zapojení nebo multidrop. S tím, že architektura Bod-Bod je nejjednodušší, ale nejdražší. Protože je nezbytné ke každému prvku vést kanál. V sériovém zapojení je počet potřebných vedených kanálů omezen, ale sdílení jednotlivých kanálů má dopad na efektivitu a komplexnost prováděných úkonů. Další dvě architektury představují kompromisní řešení. Šetří počet užívaných kanálů, ale zvětšují komplexnost celého systému[14]. Pro ilustraci je vytvořeno schéma viz obr. 1.6.



Obr. 1.6: Topologie vyskytující se ve SCADA systémech[14].

2 Užívané protokoly

PLC a RTU jednotky jsou vždy předprogramovány pro komunikaci s MTU, případně jakýmkoli jiným prvkem SCADA topologie viz výše, pomocí protokolů. Tyto protokoly jsou navrženy, tak aby byly schopny přenášet zprávy o stavu zařízení, příkazy k zařízením nebo informace od zařízení. Celý systém je postavený na hrubé ISO/OSI architektuře, respektive z ní vychází. Takže fungování SCADA protokolů je velmi podobné klasickým protokolům v rámci tohoto modelu.

V rámci SCADA systémů se využívá mnoho různých protokolů. Je jich více než 200[16]. Tato práce se však bude podrobněji zaobírat pouze čtyřmi. Pro úplnost je zde uveden přehled některých dostupných protokolů. Jedná se o proprietární i otevřené protokoly[18].

- Allen Bradley DF1,DH and DH+,
- GE Fanuc,
- Siemens Sinaut,
- IEC 60870-5,
- MODBUS,
- Omron,
- Toshiba,
- Westinghouse,
- DNP3 (Distributed Network Protocol 3),
- ICCP (Inter-Control Center Commuinications Protocol),
- UCA 2.0,
- IEC 61850,
- CAN (Control Area Networks),
- CIP (Control Information Protocol),
- DeviceNET,
- ControlNET,
- OPC (OLE for Process Control),
- Profibus.

Jak je z výše uvedeného výčtu patrné, tak o protokoly opravdu není nouze. Bohužel z toho plyne jeden nepříjemný aspekt a to je rozštěpenost systémů. Ještě před 15 lety měl každý výrobce svůj proprietární protokol. Dnes se již situace lepší. Mnozí výrobci implementuji otevřená řešení do svých zařízení, jako je právě MODBUS. Vybrané 4 protokoly jsou MODBUS, DNP3, IEC 60870-5-104, IEC 61850 (GOOSE, MMS a SMV). Protokoly jsou vybrány s ohledem na zadání této práce. Není zde tudíž řečeno, že se jedná o nejlepší možné řešení pro SCADA systémy.

2.1 MODBUS

Modbus je komunikační protokol vyvinutý Gouldem Modiconem pro využití v kontrolních systémech v 70. letech. Jeho největší výhodou je otevřenost celého protokolu a cena. Je totiž zdarma. Zaštiťuje ho organizace[1] pro Modbus. Pro komunikaci v režimu sériového přenosu může využívat hned několik standardů. Zahrnuje RS-232, RS-422 a RS-485. Protokol je možné také šířit pomocí radiového přenosu nebo po optickém vlákně[15].

Modbus je považován za spíše pomalejší ve srovnání s ostatními protokoly, ale na druhou stranu je široce akceptován mezi výrobci i uživateli. Dnes valná většina výrobců zabývajících se tímto odvětvím nabízejí produkty, které podporují jako jeden z protokolů Modbus. Modbus lze tedy považovat za standard a jeho schopnosti jsou ověřeny dlouholetou praxí. Je zde také potřeba zmínit, že Modbus má také svoji speciální verzi pro vysokorychlostní systémy Modbus Plus[15].

Modbus funguje na principu Master–Slave. Jeden master může na sebe mít navázaných 247 klientů. Čemuž i napovídá velikost identifikátoru v hlavičce, který má 1 B viz obr. 2.1. Tudíž by se pod něho teoreticky dalo adresovat maximálně 255 klientů. Ale adresa 0 je rezervována pro všesměrové zprávy. Adresace 1 až 247 jsou pro klienty. Poslední část 248 až 255 jsou rezervní a nevyužívají se. Pouze Master může inicializovat dialog. Protokol funguje na standardním schématu dotaz–odpověď. Z čehož vyplývá, že pouze Master může posílat dotazy na klienty. Dotazy mohou být zasílány pouze na jednoho klienta nebo pomocí funkce broadcast, tedy všesměrové zprávy. Všesměrová zpráva, jak už název napovídá zamíří ke všem klientským zařízením. Celá komunikace spočívá v poslání pouze jednoho rámce dotazu, ať už všesměrově či pouze na jedno zařízení a jedné odpovědi. V rámci této transakce jsou vyměněny všechny podstatné informace[15].

Transaction ID (2 Bajty)	Protocol Identifikátor (2 Bajty)	Délka (2 Bajty)	Identifikátor jednotky (1 Bajt)	Kód funkce (1 Bajt)	Data/Sub-funkce/výjimky (256 Bajtů)

Obr. 2.1: Hlavička protokolu MODBUS.

Informace, které lze přenášet mají určité značení specifické pro tento protokol. Jedná se o:

- **Diskrétní vstup (Discrete Input)** 1-bit určený pouze ke čtení (např. zapnuto–1/vypnuto–0),
- **Cívka (Coil)** 1-bit určený pro čtení i zápis (např. zapnuto–1/vypnuto–0, plus lze ovlivnit stav),

[1] Jejich stránky jsou dostupné zde: http://www.modbus.org/

- **Vstupní registr (Input Register)** 16-bitový registr sloužící pouze k čtení (např. analogový vstup),
- **Zadržovací registr (Holding Register)** 16-bitový registr sloužící ke čtení i zápisu (např. čítač, možnost nastavení i čtení jeho hodnoty).

Velice důležitou součásti tohoto protokolu jsou jeho funkce. Každá z funkcí má přiřazený identifikátor. Nejdůležitější identifikátory jsou:

01	Read Coils	Čtení jednoho nebo více bitů
02	Read Discrete Inputs	Čtení jednoho nebo více bitů
03	Read Holding Registers	Čtení jednoho nebo více 16bitových registrů
04	Read Input Registers	Čtení jednoho nebo více 16bitových registrů
05	Write Single Coil	Zápis jednoho bitu
06	Write Single Register	Zápis jednoho 16bitového registru
15	Write Multiple Coils	Zápis více bitů
16	Write Multiple Registers	Zápis více 16bitových registrů

Tab. 2.1: Přehled základních funkcí Modbus.

Jak si lze povšimnout v tabulce 2.1, tak hlavní funkce jsou Zápis/Čtení. Zápis i čtení lze provádět hromadně nebo po jednom ať už registrech nebo cívkách.

Obecné parametry Modbusu pro sériovou linku jsou tyto: rychlost 19200 baudů, 8 datových bitů a sudá parita. V Modubusu pro sériovou linku jsou definovány dva módy: Modbus RTU a Modbus ASCII. Modbus v režimu RTU je každý bajt vysílán jako jeden znak. Kontrolní součet CRC pak zajišťuje integritu zprávy. Je zde taktéž využita technika paritního bitu. Pro vysílání musí platit určitá pravidla. musí být souvislé a mezery mezi znaky nesmí být delší než 1,5 znaku. Pomlka na sběrnici, která je delší než 3,5 znaku označuje začátek a konec[15].

Díky časování a navržení protokolu, Modbus zajišťuje rychlou a spolehlivou komunikace po sběrnici. Zároveň má výhodu, že neklade příliš velké nároky na koncová/připojená zařízení. I díky těmto vlastnostem patří režim Modbusu RTU k nejvíce rozšířeným komunikačním protokolům v průmyslu. V módu ASCII je každý bajt zasílán pomocí dvojice ASCII znaků. Tento mód je mírně pomalejší než předchozí zmíněný, ale je v něm možnost zasílat znaky s mezerou až 1 sekundu. Na rozdíl od RTU mód je začátek i konec definován jinak. Dvojtečka značí začátek zprávy. Řídící sada znaků CR, LF pak značí konec. Výhodou tohoto módu je lepší čitelnost posílaných zpráv za cenu menší rychlosti. Obětování rychlosti je ale natolik silná nevýhoda, že se většinou využívá mód RTU. Zprávy jsou pak překládány na vyšších vrstvách[15].

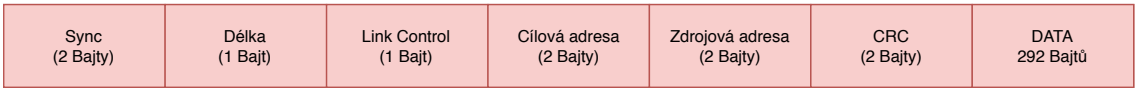
2.2 DNP 3

DNP 3 neboli Distributed Network Protocol Version 3.0 byl vyvíjen v 90. letech v Severní Americe firmou Harris Controls Division. První produkty podporující protokol se dostaly do oběhu v roce 1993[3]. Od té doby protokol nabíral na síle a podpoře více a více výrobců. Ve své podstatě se jedná o telekomunikační standard, který definuje, jak má probíhat komunikace mezi MTU a RTU jednotkami. Plus podporuje další inteligentní elektronické zařízení. Záměr při vývoji bylo dosáhnout, co největší interoperability systémů. Ve výsledku tak podporuje zařízení z mnoha koutů průmyslu. Pro představu se jedná o energetiku, ropný a plynný průmysl, vodárenský průmysl a bezpečnostní nasazení.

Na začátku byl primárně vyvíjen pro energetiku v Americe. Od té doby se však posunul a získal důvěru v dalších odvětvích průmyslu nejenom v Americe. Oblastí, kde se ale příliš neprosadil je Evropa, kde převažuje konkurenční IEC 60870. O IEC 60870 bude příští kapitola, zde je potřeba zmínit, že oproti DNP 3 má jednu velkou nevýhodu. Tou je omezení funkčnosti protokolu pouze na energetiku[3].

Podobně jako Modbus protokol je DNP3 otevřený, což mu také pomohlo získat na svoji stranu značné počty výrobců. Organice zaštiťující[2] DNP je známá pro svůj rigorózní proces zajišťování kvality. Udržují srozumitelný systém certifikátů, kde je přesně definováno, které zařízení musí mít certifikát. Další klíčovou vlastností je, že již od začátku byl vytvářen na míru SCADA systémům, takže nenese zbytečnou funkcionalitu a je naopak úzce zaměřen[3].

DNP je sice úzce zaměřen, ale to neznamená, že nemá značný výčet zajímavých funkcí zaměřených na SCADA systémy. Jako největší by se dalo považovat možnost adresovat až 65000 zařízení na jedné lince. Což je značný rozdíl oproti 247, které podporuje Modbus. Dále je přímo implementována synchronizace. Podporuje všechny topologie vyznačené na obrázku 1.6. Je zde možnost využití multi-master topologie, díky čemuž není hlavní server natolik zatížen, plus je síť distribuována, což sebou nese další benefity. Všechny zprávy posílané v rámci systému jsou dělené do více rámců, aby byla zajištěna lepší kontrola chyb ve zprávách. Jsou podporovány vše-směrové zprávy atd. Výčet by mohl pokračovat[3].

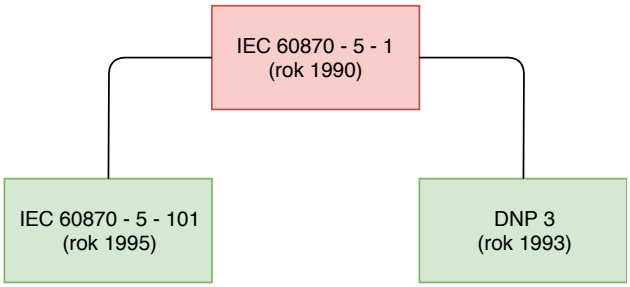


Obr. 2.2: Hlavička protokolu DNP3.

[2]Pro bližší informace o jejich činnosti lze nalézt zde: https://www.dnp.org/Default.aspx

2.3 IEC 60870-5-104

IEC 60870-5 je soubor standardů od IEC tedy International Electrotechnical Commission. Jedná se také o otevřený standard zaměřený úzce na SCADA systémy. Účelem je přenos informací a kontrola zařízení. Standard je velice detailní a definuje přesné požadavky na zařízení v jeho topologiích. Jak již bylo zmíněno, tak byl primárně vyvíjen pro energetiku. Obsahuje datové struktury, které jsou specifické pro energetiku. Lze jej však využít i jinde. Ačkoliv pro to nebyl zamýšlený. Obecně vzato je protokol používaný opravdu pouze jenom v energetice Evropských zemí. Protože ve většině dalších zemí světa je preferován standard DNP, díky jeho možnostem použití[3].



Obr. 2.3: Vývoj SCADA orientovaných protokolů.

V kontextu SCADA systémů je ale většinou zamýšlena přímo pouze část tohoto standardu. Je to IEC 0870-5-101. Tato publikace obsahuje definici pro kompletní komunikační SCADA protokol. Právě v publikaci 5-101 se objevuje kompletní definice Aplikačních datových objektů (Application data objects - APDU). APDU jsou objekty nezbytné v rámci standardu IEC 60870-5-104 pro fungování celého systému ve SCADA ekosystému. APDU jsou použitelné jak v energetice, tak v obecném využití SCADA. Zde soubor standardů prodělal vývoj, díky kterému je již možné podporovat i zařízení z jiných oblastí průmyslu. Všechny typy dat přenášených v APDU mají unikátní označení, aby nedocházelo k omylům a datové toky mohly být správně odděleny. IEC 60870-5-104 je 4. doprovodný standard, díky kterému je celkově dnes IEC 60870-5 vnímán jako univerzální protokol pro SCADA systémy. Přináší možnost přistupovat do sítě (odesílat/přijímat) za pomocí standardních přenosových protokolů definovaných pod TCP/IP. Což je značná výhoda tohoto protokolu. Jak už číslo publikace napovídá byl vydán jako poslední a to v roce 2000[3].

Podporuje pouze architektury Bod-Bod a Multidrop, ale má dokonalejší systém komunikace. Například jak Master, tak klient mohou inicializovat komunikaci, což zvyšuje efektivitu komunikace. Bohužel to platí pouze pro komunikaci Bod-Bod. Za využití architektury Multi-drop tuto schopnost ztrácí. Síť využívající IEC 60870-5 podporuje teoreticky neomezený počet zařízení v síti[3].

2.4 IEC 61850 (GOOSE, MMS a SMV)

Standard IEC 61850 je evolučním výstupem všech předchozích. Lze ho považovat za další generaci ve standardech pro automatizaci. Již při vývoji na něho byly pokládány značné požadavky. Hlavní požadavek byl, co možná nejjednodušší implementace a konfigurace. Další byl například požadavek na vysoko-rychlostní komunikaci mezi síťovými prvky, důraz na dostupnost a více-produktová interoperabilita. Velice důležitý prvek byl nově důraz na podporu bezpečnosti komunikace. Hlavní cílovou skupinou jsou prvky na nejnižší úrovni a jejich propojení. Následující tři protokoly jsou nejdůležitějšími a svými novými funkcemi definují tento standard[17].

- GOOSE (IEC Generic Object Oriented Substation Event),
- MMS (Manufacturing Messaging Specification),
- SMV (Sampled Measured Values).

Základní premisou, na které je postaven tento standard je abstrakce definice využívaných pojmů pro data a služby. Díky tomu je možné tvořit objekty a služby, které jsou nezávislé na protokolech, jež se nachází pod nebo nad jím samotným. Přináší to výhodu možnosti mapování těchto abstraktních služeb či objektů na různé další protokoly, které splňují určité dané podmínky. Definice jsou přímo obsaženy v dokumentaci. Například je zde vytvořen pojem „Logický uzel – Logical Node“, který lze „mapovat“ i na jiné protokoly, které mají ve své architektuře uzly. Pokud splňují podmínky na logical node, jež jsou dané. Díky této vlastnosti se rozšiřuje interoperabilita a kompatibilita součástek různých výrobců[17].

MMS a SMV součásti standardu se nedílně podílí právě na mapování těchto abstraktních služeb či objektů. Určitá část standardu (8.1) přesně specifikuje mapování služeb a objektů na MMS. A část zabývající se SMV definuje mapování SMV na klasickou službu Ethernet a jeho rámec. GOOSE přináší inovaci v rámci chápání konektivity v industriálních systémech. Jedná se o část standardu, která se zaměřuje na ne-spojově řízenou komunikaci (podobně jako UDP). V rámci celého ekosystému se pak využívá systém řízení síťové komunikace. Jsou podporovány VLAN a prioritizace zpráv pro GOOSE a SMV[17].

Jedna z dalších inovací tohoto standardu je, že přímo definuje, jak by si měly zařízení v topologii organizovat data. Všechny předchozí standardy definovaly pouze, jak by měly být bajty přenášeny po vodiči. Už nebylo definováno, jak by se z daty mělo zacházet v rámci aplikace. Celý systém využívá standardizovaný konfigurační jazyk. Umožňuje tím konfiguraci zařízení a definování jeho role v ekosystému pomocí XML souborů. Zajímavou funkcí zařízení fungujících pod tímto standardem je také jejich možnost se popsat. SCADA systém má tak možnost si stáhnout všechny potřebná data o 61850 zařízení přímo z něho. Může jít například o popis zařízení. Výhodou je, že toho může dosáhnout bez jakékoli manuální konfigurace[17].

3 Metodika zpracování praktické části

3.1 Výběr HW

Výběr fyzických prvků, jež budou užity v rámci práce se bude silně odrážet od typické SCADA topologie viz obr. 1.5. To znamená, že zde budou přítomny RTU/-PLC, prvky napojené na RTU/PLC (čidla, relé, atd.), router, PC a server. Přenosová struktura nebude nějak více rozebírána. Bude se jednat o standardní propojení pomocí Ethernetu. Nejenom kvůli komplikovanosti tvořit na míru šité řešení, ale i kvůli dostupnosti klasických síťových prvků. Jako router může sloužit jakékoli levnější zařízení. Například TP-LINK Archer C7 by měl naprosto dostačovat jako vhodný kandidát. Vzhledem k faktu, že se jedná o malou topologii, tak není potřeba více vstupů integrovaného přepínače. Přímo na router bude napojeno PC s bežícím programem pro HMI a server se SCADA systémem. Pak na něho bude napojený přepínač s nízkoúrovňovými prvky. Plus je zde možnost zapojit data loogger nebo recovery server.

Důležitou otázkou je zda SCADA systém bude provozován na dedikovaném serveru nebo na PC pomocí virtualizace. Vzhledem k požadavku na mobilitu a přenositelnost systému, jež byla zadána. Je nezbytné, aby systém byl virtualizován na klasickém PC. Vytvořený virtuální disk bude umožňovat jednoduchou portabilitu a možnost spustit systém téměř kdekoli, bez toho aniž by byla potřeba větší konfigurace. V rámci systému pro virtualizaci bude alokováno SCADA systému dvě logická jádra na CPU od firmy AMD a 4 GB paměti pro hladký běh aplikací. Zmíněné parametry by měly na začátek plně dostačovat. Výhodou virtualizovaného systému je také fakt, že pokud by systému začala docházet paměť, případně výkon CPU, je zde možnost téměř okamžitě navýšit dané parametry.

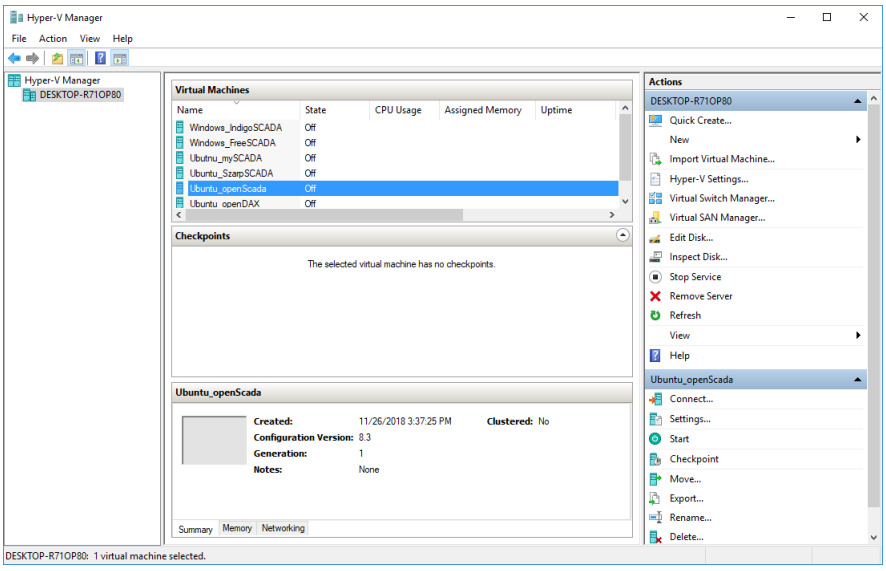
PC by teoreticky měly být dvě. První zmíněné již výše v textu by mělo obsahovat poměrně silný HW pro možnost budoucího růstu a svižnost systému. Tedy nějaký alespoň 8-jádrový systém s SMT nebo Hyperthredingem a 16 GB RAM. Druhý PC již nemusí mít takový výkon vzhledem k tomu, že na něm poběží pouze HMI. V praxi se samozřejmě může jednat o jeden PC a druhý virtualizovaný systém, ale to již bude záviset na implementaci, která bude rozebrána, až v následujících kapitolách. Potenciálně by se zde mohl objevit zálohovací server tzn. síťové úložiště s právě aktuální konfigurací.

Posledním prvkem jsou RTU/PLC. Zařízení přímo určená na tuto roli se pohybují v rámci tisíců až deseti-tisíců korun. Proto v rámci práce nebudou použity, ale je zde možnost je samozřejmě s jednoduchostí použít, pokud to bude nezbytné. Jako obejití finanační stránky věci, jsou zvoleny Raspberry PI zařízení na nichž bude instalován SW na simulování PLC/RTU prvků.

3.2 Volba SW

Hlavní část volby softwaru, tedy SCADA softwaru bude rozebrána až v další kapitole. Zde jsou rozebrány pouze obecné součásti systému. Nejdůležitějším prvkem ze SW jsou operační systémy. Zde byl vybrán jako operační systém Ubuntu ve verzi 18. Windows byl použit pro hostování systému Ubuntu nebo popřípadě, pokud některý ze SCADA systémů byl určen pouze pro operační systém od Microsoftu. V určitých částech práce bude Raspberry PI pouze virtuálním zařízením. K tomu poslouží operační systém Raspbian x86, který naprosto spolehlivě duplikuje funkce Raspberry PI. Klient, který bude virtualizované systémy hostovat bude Hyper-V, vzhledem k jeho výchozí přítomnosti v systému Windows. Jeho přenositelnost je také velice dobrá, vhledem k formátu VHDX, který umožňuje klonované disky přenášet.

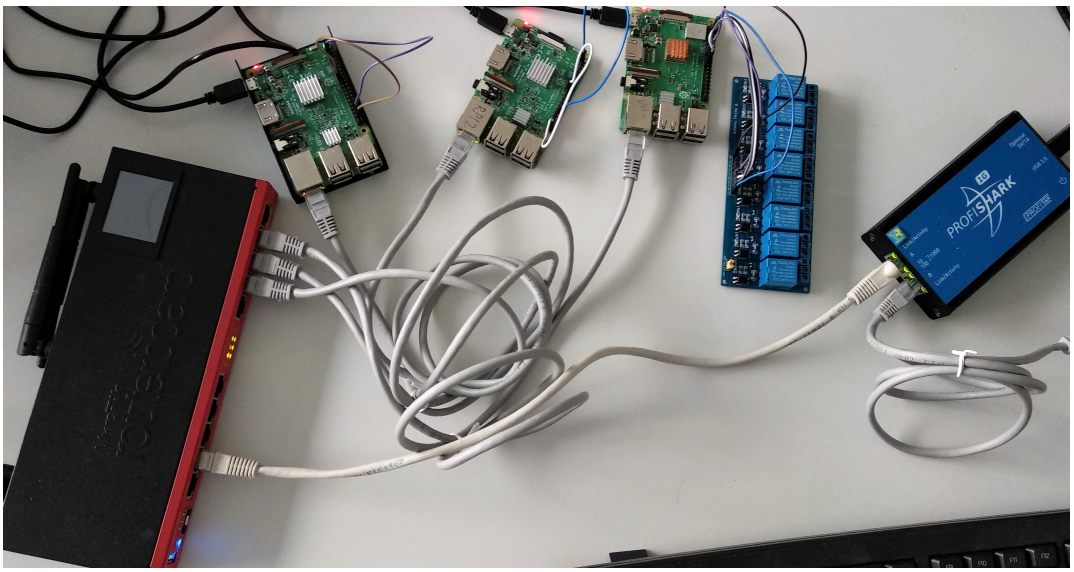
Z pomocného softwaru bude využit Wireshark pro sledování komunikace. Wireshark jako odzkoušený software by měl poskytnout detailní informace o probíhající komunikaci. Případně později i pomoci při diagnostikování problému mezi zařízeními. Pro simulování a práci s protokolem Modbus je využit program Modbus Slave. Jedná se o velmi jednoduchý, ale zároveň velmi propracovaný program. Umí simulovat až 32 klientských zařízení najednou[19]. Podobný program bude využit i v případě testování funkčnosti implementace DNP 3 protokolu. Jedná se Opendnp3 Simulator, jež je součástí iniciativy Opendnp3. Oproti předchozímu zmíněnému má možnost i emulovat server tedy řídící jednotku. Je silně přizpůsobitelný a konfigurovatelný[20]. Z hlediska obecně použitého SW jsou již všechny důležité programy zmíněny a výběr SCADA softwaru bude pokračovat v další kapitole.



Obr. 3.1: Rozhraní virtualizačního klienta Hyper-V od Windows.

3.3 Laboratorní přípravek IEC 61850

V rámci zadání této práce je seznámení se s přípravkem, kde je již IEC 61850 implementován. Implementace proběhla v rámci diplomové práce[21]. Prvek figuruje jako slave/klientské zařízení. Master stanici mu dělá PC s běžící knihovnou simulující SCADA prostředí. Komunikace probíhá po Ethernetu a pak po sériové lince ke konečným prvkům topologie. PLC je nahrazeno RaspberryPi, což umožňuje jistou flexibilitu. Minimálně rozhraní Ethernetu nechybí, případně i USB. Vše je napojeno přes přepínač na PC, čehož bude využito později v této práci. Protokol úspěšně implementovaný je tedy IEC 61850. Všechna komunikace je tudíž umožněna na základě tohoto protokolu. Pro více informací o zapojení či konfiguraci všech prvků je dostupná publikace již zmíněné diplomové práce[21].



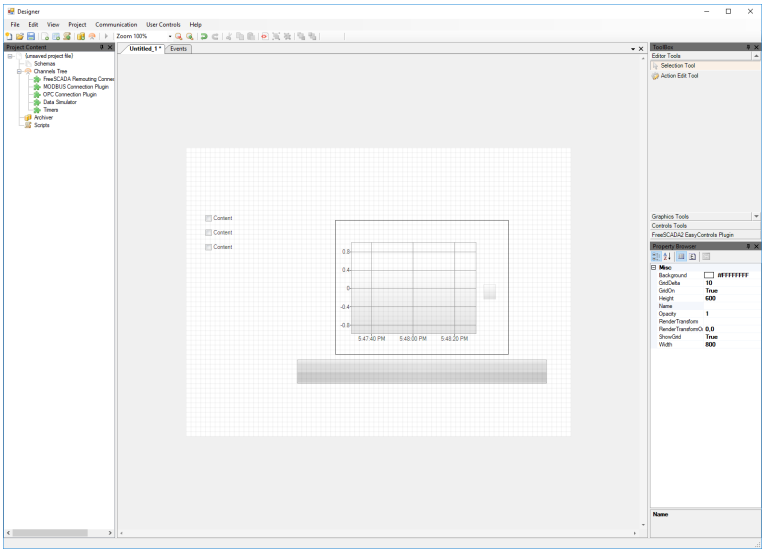
Obr. 3.2: Pracoviště s přípravkem IEC 61850.

4 Nástroje pro vizualizaci dat

4.1 Otevřené řešení (Open-source software)

4.1.1 Free SCADA 2

Free SCADA 2 je nástupcem první iterace tohoto systému z roku 2006. Původní Free SCADA byl jedním z prvních opravdu zdarma open source softwarů pro SCADA systémy. Bohužel je již bez podpory nebo pouze minimální a to se týká i novější iterace. Tato SCADA v nejnovější verzi byla vydána naposledy v roce 2009. Není se ostatně čemu divit, vyvíjel ji pouze jeden člověk Michael Tutin[22]. Ten s podporou skončil právě v roce 2009. Z hlediska podpory systému je tedy na tom hůře než „konkurenti“. Poslední systém mající nativní podporu je Windows Vista. Ačkoliv při testování se systém bez problému spustil i na Windows 10. Systémy na bázi Linuxu nejsou podporovány vůbec. Z popisu autora o systému: FreeSCADA je open source SCADA systém pro Windows 2000/XP/Vista. Poskytuje uživateli flexibilní nástroje pro vizualizaci a interaktivní kontrolu jakéhokoliv průmyslového procesu.

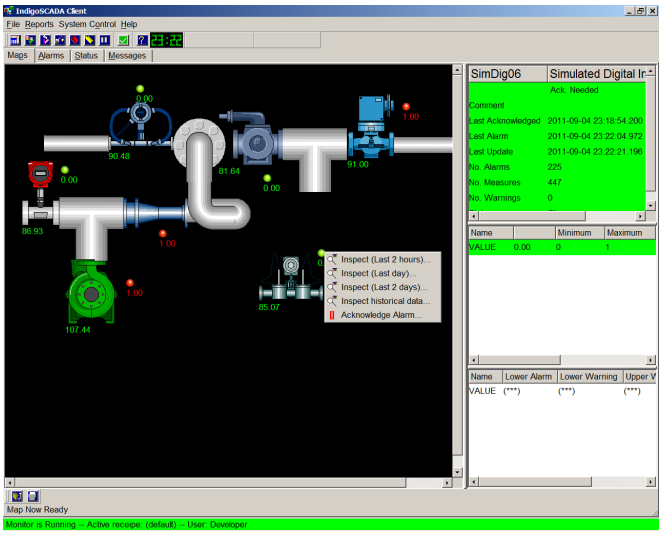


Obr. 4.1: UI designeru FreeSCADA 2.

Systém je uživatelsky dosti nepřizpůsobivý, protože neexistuje žádná detailní dokumentace. Uživatel musí na všechny prvky ovládání přijít sám. A vzhledem k velikosti projektu neexistují žádná fóra ani helpdesk. Na druhou stranu, prvky jsou uloženy docela logicky a je snazší se zorientovat než se zdá. Na první pohled je patrné z Obrázku 4.1, že UI je provedeno opravdu jednoduše a žádná nápověda neexistuje. I přesto autor tohoto textu byl schopen v relativně krátkém čase, vytvořit funkční UI na obrázku.

4.1.2 IndigoSCADA

IndigoSCADA je jak sám sebe prezentuje „Small footprint SCADA system“, což znamená, že je malý a výkonově nenáročný. Rozdíly ale nebyly pozorovány oproti ostatním SCADA systémům, takže je potřeba brát toto tvrzení se skepticismem. Ačkoli je možné, že nějaké rozdíly se projeví ve velkých topologiích. IndigoSCADA je psána v C/C++ 98 což by též mohlo přispět k lepšímu chodu celého systému. Je zde potřeba dávat pozor na určitou věc. Protože na-rozdíl od FreeSCADA systému tento není plně opensource. Ačkoliv se tak prezentuje, tak ne všechny součásti jsou opravdu opensource. Nicméně valná většina je[23]. Pro některé uživatele by se to mohlo stát rozhodujícím faktorem, proč tuto platformu nevyužít. Pokud by uživatel totiž měl zájem o využití již implementovaného standardu IEC 60870-5, tak nemůže nebo pouze za poplatek.



Obr. 4.2: UI klientu IndigoSCADA.

Na druhou stranu valná většina dalších protokolů/standardů je podporováno pod licencí GPL, tudíž jsou open-source a zdarma. Včetně standardů jako DNP3 nebo Modbus, takže by uživatel neměl mít problém najít alternativu k 60870-5 a mít tak systém bez problému implementovaný. Systém funguje lépe na operačním systému od firmy Windows, protože na něho byla primárně zamýšlena. Linux podpora je až nyní dodělávána a nachází se ve verzi Alpha testování.

Z hlediska funkcionality je ale opravdu těžké cokoliv tomuto systému vytknout. Na systému Windows funguje bez problému. Podporuje mnoho funkcí jako třeba jednoduché zálohy, více-displejové HMI, grafické rozhraní prezentující real-time data atd. UI sice vypadá spartánsky viz obr. 4.2, ale poskytuje mnoho funkcí a není problém najít jakoukoliv nápovědu. Plus je systém stále ve vývoji a stojí za ním společnost EnsSCADA, která zajišťuje podporu.

4.1.3 openDAX

O systému OpenDAX je velice obtížné získat jakékoliv informace. Původní vývojář za tímto projektem byla společnost PetraSoft Inc. Dnes jsou již stránky mimo provoz a společnost se zaobírá jiným typem činnosti. OpenDAX je zde tedy zahrnut pouze z důvodu úplnosti a vzhledem k tomu, že byl jmenován v zadání této práce. Jinak historicky byly vždy podporovány pouze operační systémy na bázi Linuxu. Jednalo se o opensource systém, který se zaměřoval na modularitu. Podporoval Modbus a centralizovanou real-time databázi, takže z hlediska funkcionality na tom také nebyl nejlépe, protože podporoval opravdu pouze základ. Instalace systému ze zdrojového kódu byla velmi náročná a pouze částečně úspěšná. Protože se systém nedokázal plně zkompilovat a tudíž nefungoval. Z praktického hlediska je možné tento projekt považovat za nepoužitelný.

4.1.4 S.E.E.R. 2

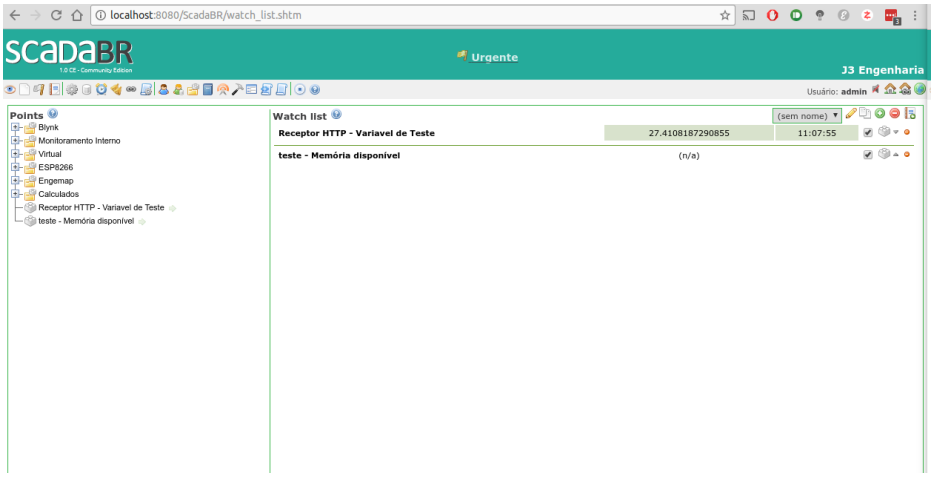
S.E.E.R. 2 je již plně rozvinutá platforma. Jedná se o nejenom SCADA systém, ale je zde implementován také historian a agregovaný systém pro analýzu. Je vyvíjen pod licencí GPL, takže je plně opensource. Z operačních systémů je primárně podporován Linux. Ale poslední aktualizace, co byla vydána přidala podporu systémů Windows se zaměřením na jejich server edice. Systém jako celek bohužel přestal být ve vývoji od roku 2016. Od té doby již nevyšla žádná aktualizace a není důvod se domnívat, že se to změní. S.E.E.R. 2 je přeložený dokonce do 4 jazyků. Jeho hlavní výhodou je webové rozhraní, na kterém staví. Celá platforma je napsána v jazyku PHP[25]. Instalace, jako taková je obtížná, ale zvládnutelná.

4.1.5 SCADA Process Viewer

Tento projekt je opět velmi malý. Podporovaný pouze jedním člověkem Sergey Maslovem. Vývoj je již opět u konce ve verzi pre-alpha 0.01. Bohužel nebylo možné ani software nainstalovat. Vypadá to na problém s kompatibilitou. Projekt byl totiž naposled aktualizován v roce 2013. Jediným podporovaným systémem je Windows, ale již není specifikované jaké verze. Projekt byl realizován opět pod licencí GPL, tedy opensource. Je naprogramován v jazyku Pascal[26].

4.1.6 ScadaBR

ScadaBR je již velmi rozvinutý systém. Má silnou podporu od svého výrobce se stejným jménem. Bohužel hodně z dostupných materiálů je pouze portugalsky, vzhledem k tomu, že systém je zaměřen hlavně na publikum z portugalsky mluvících zemí. Naštěstí samotný systém je v angličtině. Z operačních systémů je podporován pouze Windows, ale ve vývoji je nová verze ScadaBR, která má ještě nespecifikované podporované systémy. Je zde velká pravděpodobnost, že zahrne Linux i mobilní platformy. Systém jako takový je již dlouholetě používán ve všech odvětvích průmyslu včetně chemiček, inteligentních budov atd. Tento systém má opravdu rozvinutou podporu včetně nabízených vzdělávacích kurzů, které by měly všem pomoci k rychlejšímu osvojení platformy. Bohužel jsou všechny lokalizované pouze do portugalštiny.

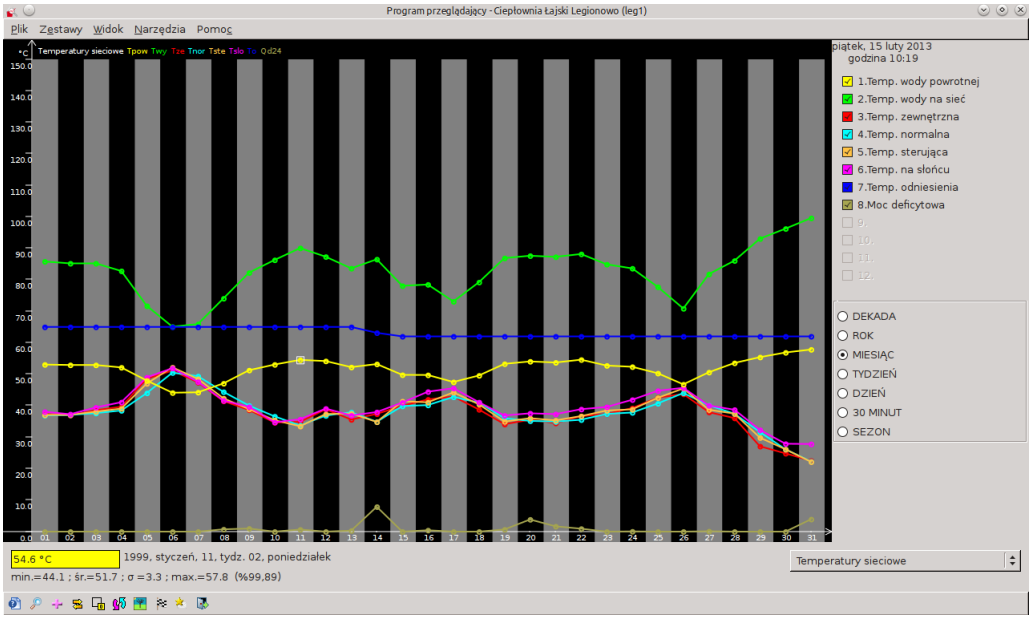


Obr. 4.3: UI systému ScadaBR.

Instalace na systému Windows je poměrně složitá, protože je nezbytné systém propojit s databází MySQL. K tomu je potřeba zprovoznit Apache Tomacat na localhostu. To vše je ale vykoupeno tím, že celá aplikace je webová, což přináší značné benefity. Například, že je aplikace velice mobilní. Je možnost k ní přistupovat z jakéhokoliv zařízení. UI je velice simplicistické viz obr. 4.3. Celkově systém působí vyzrále a přehledně. Jsou zde přítomny všechny prvky a s překladačem se dá dojít i k rozumným návodům na použití, pokud se nebude počítat základní manuál v angličtině. Je podporováno až 20 protokolů včetně všech důležitých tzn. různé iterace protokolu Modbus (Modbus TCP/IP, serial atd.), DNP 3, IEC 60870-5. Je zde podporována implementace jednoduchého a přehledného HMI. Poskytuje možnost využít služeb Dataloggeru/Historianu. Jako všechny ostatní zmíněné systémy je i tento zdarma a opensource. Placená je až podpora. Je zde potřeba zmínit, že se jedná stále o silně vyvíjený software, který přidává stále novou funkcionalitu. Bohužel lokalizace stále pokulhává[27].

4.1.7 Szarp

Szarp je stále vyvíjený systém již od roku 1991 firmou Praterm. Vyvíjený je pod licencí GPL, takže se jedná o opensource a je zdarma. Jedná se o polský systém využívaný a testovaný v místních továrnách. Ale s lokalizací není problém a je zaměřený na anglicky mluvící publikum, alespoň návody ano. Ze systémů je plně podporován Linux a Windows pouze částečně. Na systémy postavené na platformě Linux lze nainstalovat celý balíček Szarpu. Instalace je poměrně jednoduchá za pomocí repositářů. Důležité je, že obsahuje dobrou dokumentaci pro řešení nastalých problémů. Na Windows lze nainstalovat pouze klienta, takže nelze říci, že by byl plně podporován, ale například pro kontrolu je dostačující. Instalátor je však v polštině, což značně znepříjemňuje instalaci. Z toho plyne, že systém je primárně zaměřen na Linux, přesněji na Debian a Ubuntu plus Raspbian, jak sami autoři deklarují. Poslední vydána verze je 3.1. Obsahuje možnost integrovat prohlížení procesů do Firefoxu, kreslení schémat pomocí Draw3. Ze známých protokolů je však podporován pouze Modbus RTU/TCP. Dále jsou podporovány proprietární protokoly Praterm ZET/Sterkom PLC.

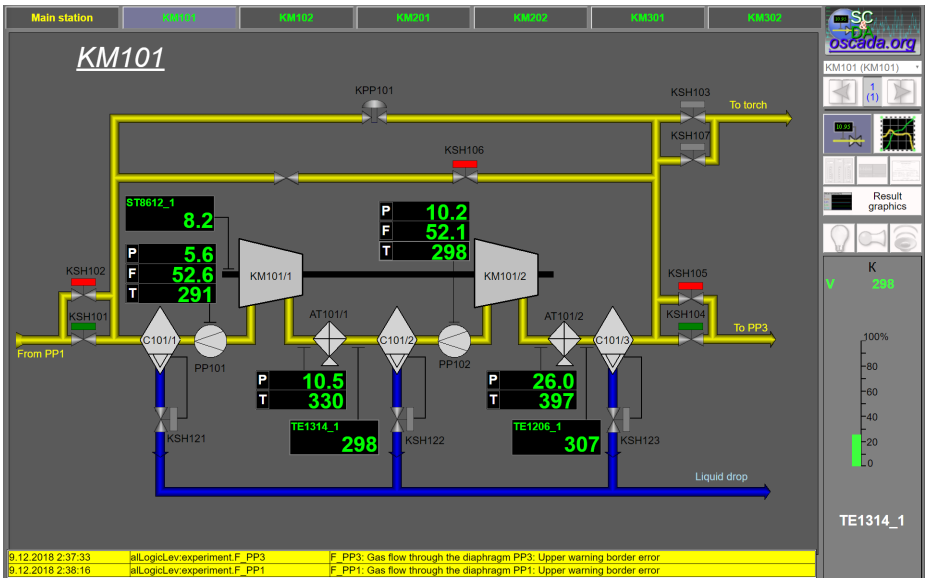


Obr. 4.4: UI systému SzarpSCADA.

Na UI je již vidět zub času, ale je přehledné viz obr. 4.4. Systém obecně nepůsobí moderně oproti ostatním SCADA systémům, které například dokáží běžet v prohlížeči. V rámci programu se nachází implementace nového protokolu Danfoss FC. Bohužel protokol není podporován žádným jiným velkým výrobcem. K protokolu je však dostupná detaliní dokumentace a je možné, že se uchytí v budoucnu. Pro začátek to bohužel výhoda není[28].

4.1.8 openSCADA

OpenSCADA je nezávislý projekt ukrajinského původu. Jedná se o otevřenou implementaci SCADA systému i s HMI. Je kompletně zdarma. Dokumentace je na solidní úrovni v anglickém a primárně ruském jazyce. V současné době se čeká na vydání verze 1.0, která sebou přinese kompatibilitu se systémy Windows. Nyní je dostupná pouze na systémech Linux, ale dá se říci, že na všech. Jenom namátkou se jedná například o Debian, Ubuntu, Fedora, CentOS, atd. Instalační proces je velice detailně popsaný přímo na oficiálních stránkách. Tudíž instalace byla poměrně jednoduchá za použití repositářů.

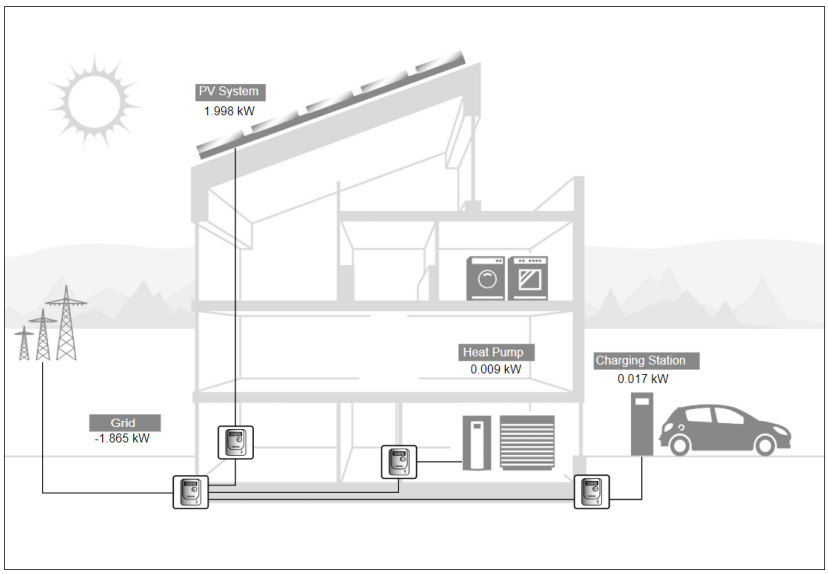


Obr. 4.5: UI procesu v systému openSCADA.

OpenSCADA funguje na tzv. modulární architektuře. To znamená, že pokud uživatel nemá zájem na provozování určitého protokolu nebo rozhraní, tak je prostě neimplementuje do své základní SCADA instalace. Což sebou nese řadu výhod. Ve svém základu je openSCADA velice nenáročný program. Záleží až pak na implementaci nakolik bude široká. V rámci protokolů zde je podporovaný Modbus, IEC 60870-5, OPC a další. Dokonce je zde připravená šablona pro uživatele, na základě které si může tvořit vlastní implementace protokolů. Pokud je to nezbytné. UI konfigurátoru je velice obsáhlé. Uživateli trvá než pochopí všechny souvislosti. Naštěstí zde je podpora komunity dost silná, takže je schopná pomoci. V rámci grafického rozhraní pro bežící procesu je vše, co uživatel může potřebovat a je přehledné. Hlavní myšlenka celého projektu je rozšiřitelnost, flexibilita a otevřenost. Přistupovat k systému lze multiplatformně. Což znamená, že konfigurace musí proběhnout z Linuxu, ale kontrola již může probíhat z jakéhokoliv zařízení. Včetně mobilních zařízení a webových prohlížečů[31].

4.1.9 openMUC

OpenMUC je další zástupce SCADA systémů. Vzniká v sousedním Německu pod patronátem Ministerstva obchodu a energetiky. Stojí za ním skupina Smart Grid ICT sídlící ve Fraunhoferově institutu pro solární energetiku[32]. Hlavním zaměřením tohoto projektu jsou tudíž technologie zaměřené na solární energetiku a energetiku obecně. Jedná se například o fotovoltaické systémy, baterie, tepelná čerpadla, elektrárny a elektro-automobilita. OpenMUC systém je tedy primárně zaměřený na energetiku, ale vzhledem k podpoře mnoha protokolů na ni není limitován.



Obr. 4.6: Příklad openMUC vizualizace.

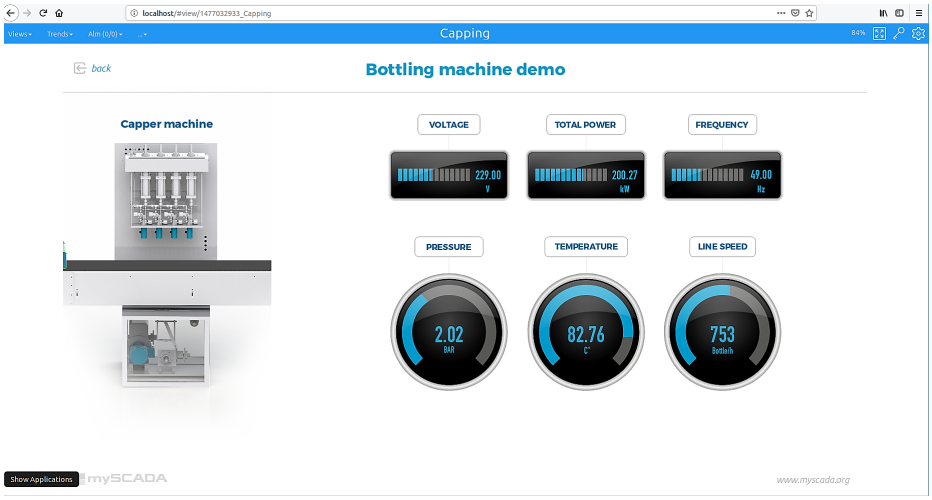
Tento SCADA systém není považován za hotový program, ale spíše za framework ulehčující vývoj velkých SCADA systémů. Primárním vývojovým jazykem je JAVA, což sebou nese značnou výhodu kompatibility na mnoha systémech. Systém je tudíž možné spustit na jakémkoli operačním systému. Hlavní branou pro přistupování a nastavování systému je jeho webová aplikace. Spuštění vyžaduje určitou znalost jednotlivých systémů, ale nedá se označit za náročné. Nastavování jednotlivých protokolů je už otázka pokročilejšího studia fungování frameworku. Podpora protokolů již byla zmíněna. Je poměrně značná, což zahrnuje např. ModBus, IEC 61850/60870-5-104/62056-21, DLSM, KNX, atd. Přístup k fungování celého framworku je ne nepodobný tomu v systému openSCADA. Je zde také kladen velký důraz na modularitu a přizpůsobení si vlastních aplikací. Systém je vyvíjený pod licencí GPL, což přináší otevřenost kódu veřejnosti. Na obrázku je například vizualizace z demo aplikace vytvořené pod openMUC. OpenMUC framework je nenáročný na využití systémových prostředků, a proto pod ním může fungovat téměř neomezené množství aplikací[32].

4.2 Uzavřené řešení (Proprietary software)

4.2.1 mySCADA

MySCADA je uzavřené a placené řešení vyvíjené taktéž stejnojmennou firmou. Je to první SCADA systém, který je zde zmíněn a nemá otevřený kód. To znamená, že je nemožné si tento software jakkoliv upravit na úrovni kódu. Na druhou stranu přináší mnoho výhod, které ostatní softwary nemohou poskytnout. Například pro české uživatele je markantní výhoda, že firma, která vyvíjí mySCADA sídlí v Česku. MySCADA má tedy pro česky mluvící publikum silnou výhodu přímo české lokalizace. Jedná se o universální řešení pro HMI i SCADA systém pro všechna průmyslová odvětví. Patří sem například ropný průmysl, energetika, vodárenský průmysl atd. Software se sice dá sehnat zadarmo přímo ze stránek výrobce, ale každé 2 hodiny se restartuje. Což by se dalo považovat za určité demo. Takže hlavní nevýhoda je nezbytnost si software zakoupit. Ačkoliv v ceně je podpora a řešení na míru, každé společnosti.

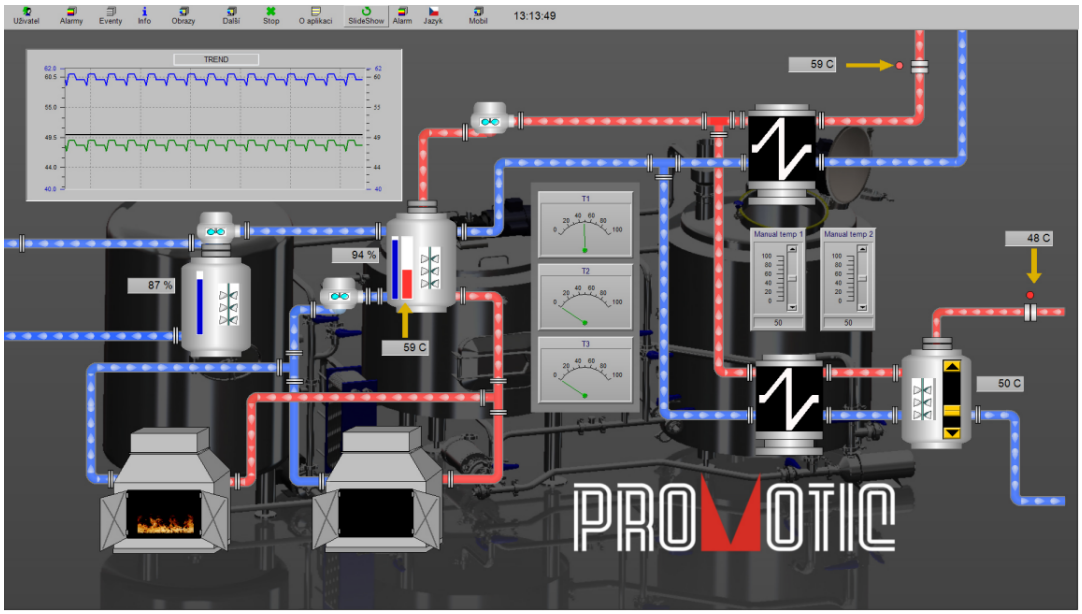
Jedna z velkých výhod stabilního zázemí společnosti, která software vyvíjí je podpora, která je u tohoto produktu na špičkové úrovni. Což se například odráží v počtu podporovaných systémů. Jsou podporovány operační systémy na bázi Linuxu, Windows i Mac OS X. Jedná se tedy o první systém, který platformu od Apple podporuje. Na poli protokolů lze nalézt mnohé. ModBus je plně podporován jak v módu TCP, tak serial. Dále zde jsou proprietární protokoly od firmy Siemens a Melsec atd. Jako každý moderní systém funguje mySCADA ve webovém prohlížeči, takže je možné k ní kdykoli přistupovat. UI je velice přehledné a moderně vypadající[29].



Obr. 4.7: UI systému mySCADA.

4.2.2 PROMOTIC

Promotic je taktéž vyvíjen jako proprietární software a není tudíž opensource ani zdarma. Respektive je zdarma jeho demo režim, který poskytuje plnou funkcionalitu do 30 zařízení, pak je již nezbytné si software zakoupit. Je vyvíjen českou firmou Microsys s.r.o sídlící v Ostravě. Jejich prezentace je tudíž ryze česká. Firma má navázané vztahy se vzdělávacími institucemi. Promotic je již 27 let ve vývoji a jedná se tudíž o velmi vyspělý systém. Dokonce byl zvolen softwarem roku 2016 pod veletrhem AMPER 2017 v Brně. Z hlediska podpory protoklů je na tom Promotic velmi dobře a podporuje Modbus i IEC 60870-5, plus mnohé další. Lokalizace je velmi rozsáhlá. V konfiguračním módu je možnost si zvolit češtinu, angličtinu a polštinu. V runtime módu pak mnoho dalších.



Obr. 4.8: Příklad UI systému PROMOTIC[30].

Promotic se nyní nachází ve verzi 9 z roku 2018 a stále se na něm pracuje, přibývají lokalizace i funkcionalita. Použití nalezne například v energetice, chemickém průmyslu, v hutích, atd. Podpora a dokumentace jsou dobře zvládnuté. Z operačních systémů je podporován pouze Windows. Je kompatibilní s různými druhy databází včetně MySQL, dBASE a Access. Vytvoření projektu je jednoduché a UI je přehledné. Nepůsobí sice tak moderním dojmem jako mySCADA, ale i tak je velice dobře zvládnuté[30].

4.3 Shrnutí

V předchozích kapitolách bylo představeno 11 systémů. Každý se svými specifickými výhodami a problémy. Pro účely této práce je ale potřeba vybrat jeden. Jeden, který bude splňovat určité podmínky. Základní podmínkou pro začínající budování nového prostředí pro SCADA systém je určitě cenová dostupnost. Jedním z hlavních ukazatelů tak bude cena. Další důležitý aspekt jsou možnosti úpravy daného softwaru. Takže pokud je software opensource nebo pokud má nějakou formu zjednodušení, jak si upravit podle sebe, tak splňuje tuto podmínku. Dalším důležitým aspektem je určitě dlouho trvající podpora, která bohužel není u všech systémů na takové úrovni, aby byla vyhovující. Na základě těchto, ale i dalších parametrů bude vytvořena přehledná tabulka níže.

SCADA systémy	Otevřenost systému	Cena	Dostupnost	Podpora	Linux	Windows	ModBus	Rošiřitelnost
Free Scada 2								
IndigoSCADA								
openDAX								
S.E.E.R. 2								
SCADA Process Viewer								
ScadaBR								
Szarp								
openSCADA								
mySCADA								
PROMOTIC								
openMUC								

vyhovující částečně vyhovující nevyhovující

Obr. 4.9: Srovnání SCADA systémů.

Vzhledem k faktu, že požadavek na rozšiřitelnost a modifikovatelnost byl vyjádřen již v zadání, tak není možné využít ani jedno z komerčních řešeních. Ačkoliv obě řešení jsou na vysoké úrovni a jejich dokumentace detailní. Normálně by bylo potřeba silně zvažovat jejich výhody. Už jenom pro důvod, že se jedná o tzv. řešení na klíč. Bohužel však musejí být vyřazena z této práce, právě kvůli řešení na klíč. Na poli Opensource je již větší volnost. Zde každý ze systémů umožňuje si svůj systém v nějaké míře upravit. Na druhou stranu problémy, které netrápí uzavřené verze a jsou tam dobře zvládnuty, jsou zde velmi viditelné. Majorita nejenom zmíněných opensource systémů má značné problémy s udržováním podpory svých systémů. U jednoho z nich dokonce naprosto přestaly fungovat stránky a zdrojový kód je téměř nevyhledatelný. Další mají podporu pouze v jazyce, který není angličtinou.

Z tabulky je jasné, že problémy mají všechny systémy v nějakém aspektu. Nejlepší tři systémy se na první pohled zdají IndigoSCADA, openSCADA a openMUC. Všechny tři mají mírně odlišný přístup. Zatímco IndigoSCADA má mnoho funkcionalit a je velmi robustním systém, openSCADA a openMUC volí modulární přístup.

Oba dva přístupy mají svá pro i proti. Ale pro tuto práci je mnohem podstatnější modularita systémů openSCADA a openMUC. Protože kdyby byla nezbytnost implementovat protokol, který není výchozí pro IndigoSCADA (což není pravděpodobné), tak by to byl značný problém. Pro openSCADA i openMUC systémy je na internetu již dostatek "modulů"a všechny se dají upravovat, takže by to neměl být až takový problém. Je zde možnost nalézt právě hledaný modul a přidat ho. Respektive s nějakou konfigurací by toho neměl být problém dosáhnout. Protože právě přístup openSCADA a openMUC systému vyžaduje větší pochopení. V IndigoSCADA je vše již před-implementované, což by mělo ulehčit práci, ale zároveň znesnadňuje modifikaci. Proto je pro budoucí použití vhodnější využití openSCADA nebo openMUC systému. Z čehož vyplývá, že hlavní rozhodnutí, který systém bude použit, je mezi openMUC a openSCADA. Zde se začínají projevovat nuance například v podpoře. Nevýhody, které jsou u openSCADA systému přetrvávající je absence verze pro Windows a převážně rusky psaná fóra. Tento problém se u openMUC neobjevuje. Vše je primárně psané anglicky. Bohužel i openMUC není bezchybný z hlediska podpory. Mnohé informace o konfiguraci a využití systému chybí. Nicméně openMUC je zvolen pro další pokračování této práce, díky své větší přístupnosti a modernějšímu návrhu fungování. Ale v reflekci na předchozí zmíněné bude i částečným cílem této práce více zpřístupnit možnosti konfigurace openMUC systému. Svým způsobem zdokumentovat nasazení tohoto frameworku.

5 Návrh kybernetického prostředí

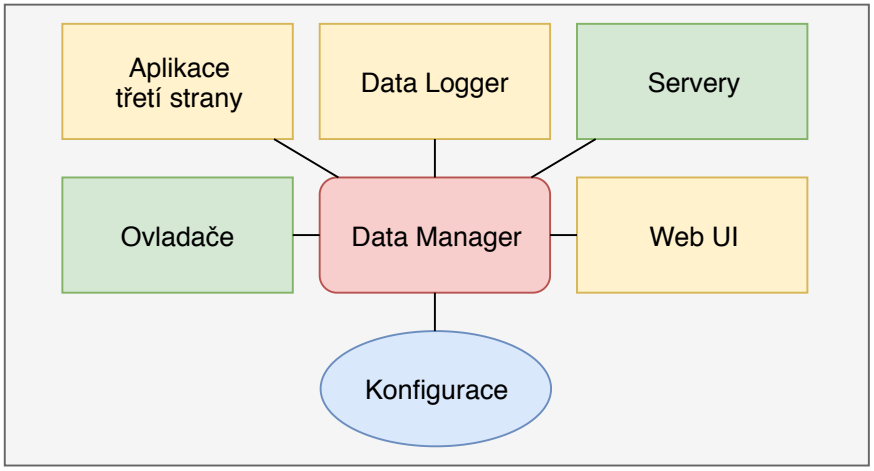
V této části je nastíněno, jak by výsledné kybernetické prostředí mohlo fungovat a vypadat. Nedílnou součástí SCADA kybernetického prostředí je samotný systém SCADA. Výběru SCADA systému se tato práce věnovala v předchozích kapitolách. Zde budou uvedeny například moduly použité pro výsledné prostředí. Další položkou budou využité protokoly/standardy, pomocí kterých se informace budou po síti šířit a v neposlední řádě se jedná o samotné prvky. Jedním z nich bude například již zmíněný laboratorní přípravek s implementovaným IEC 61850. Nebo se může jednat o virtualizované prvky systému, kterým bude ještě věnován prostor. Hlavním účelem navrženého prostředí by mělo být spojení HMI, SCADA systému a RTU/PLC jednotky.

Ze zadání je požadavek na snadnou rozšiřitelnost a modifikovatelnost celého prostředí. Vhledem k tomuto faktu byl vybrán openMUC. Majoritní část práce bude probíhat právě v tomto prostředí. Bude zde potřeba implementovat ty správné ovladače, aby řádně celý systém fungoval. Jedna ze zásad při práci s celým rozhraním bude ho nepřetěžovat zbytečnými ovladači. Cíl je dosáhnout, co možná „nejlehčího“ systému, ale zároveň aby měl všechnu funkcionalitu nezbytnou pro správný chod. Zde se bude odrážet požadavek zadání. Protože prostředí bude současně poměrně rychlé na zorientování, vzhledem k menšímu počtu ovladačů. Nicméně pokud v budoucnu bude někdo chtít přidat ovladač pro KNX standard, tak bude mít tu možnost dosáhnout svého cíle rychleji. Vzhledem k tomu, že kostra bude existovat. Modulární architektura tedy bude zajišťovat jednoduchou rozšiřitelnost. Z hlediska modifikovatelnosti je potřeba již pochopení, jak ovladače fungují v rámci openMUC systému. Ale důležité je, že modifikovatelnost je umožněna.

Jak již bylo řečeno ovladače budou nedílnou součástí této práce, proto zde budou některé z nich zmíněny. OpenMUC si přímo definuje, jak každý ovladač musí být napsán, aby mohl být implementován. Což klade vysoké nároky na toho, kdo ovladač implementuje. Ve středu jádra openMUC je Data manager[33]. Data manager musí být zapojen vždy a je nedílnou součástí celého systému. Přiřazuje priority procesům, kontroluje integritu ovladačů a jejich správnou funkčnost. Dá se říci, že je mozkem celého systému. Na něho jsou napojeny jednotlivé sekce, skládající se z logický celků viz obr. 5.1. Nejdůležitější součástí Data manageru je soubor Konfigurace tzn. Configuration File (zvýrazněný modře), ze kterého později budou i brány výpisy pro demonstraci nastavení určitých prvků.

Prvky zvýrazněné žlutě představují moduly, kterými se tato práce bude zaobírat pouze okrajově. Minimálně dva ze tří musí být využity, ale nebudou v nich dělány žádné úpravy, protože to není nezbytné. Modulem Aplikace třetí strany pro openMUC se tato práce nebude zabývat příliš zaobírat. Jenom pro úplnost, modul umož-

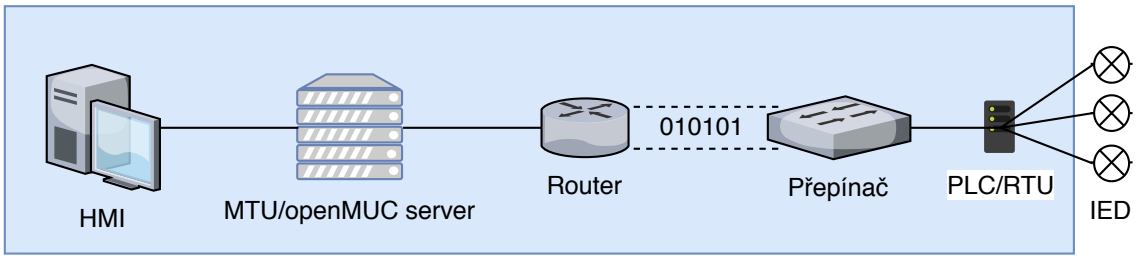
ňuje implementovat jakoukoli uživatelskou aplikaci. Zde se projevuje otevřenost celé platformy, protože pokud by uživateli nestačily již implementované aplikace, může si je vytvořit. Například je možné si vytvořit svůj vlastní Data Logger nebo speciální modul akvizice dat přes Internet atd. Další součástí je Data Logger. Slouží k zpracování dat. Data mohou být ukládána v několika formátech. Výchozí formáty jsou ASCII a SlotsDB. Pokud si však uživatel přeje je možné přidat jakýkoli další formát. Modulu WebUI je věnován prostor později.



Obr. 5.1: Modulární architektura openMUC.

Nejdůležitější prvky pro tuto práci jsou zvýrazněny zeleně. Jedná se o moduly pro Ovladače a Servery. Prvek Ovladače obsahuje všechny protokoly, které si uživatel žádá používat. Nachází se zde mnoho modulů pro různá rozhraní a standardy[33]. Na úplný začátek bude důležité vůbec zprovoznit celou topologii viz obr. 5.2. K tomu bude použit ovladač ModBus. Jako legacy protokol by neměl být velký problém zprovoznit všechny prvky sítě. Vzhledem k tomu, že je zde plně zaintegrovaný, tak nejsou očekávány žádné problémy. Modul Servery bude využit pro přistupování k openMUC přes síť. Bude navržena testovací topologie, kdy bude primárním zaměřením zjistit, jak dobře se přistupuje k openMUC z veřejné sítě.

Tím zbývá poslední prvek a tím je Web UI. Web UI silně souvisí s HMI. Uživatelské rozhraní včetně ovládacích prvků je již přehledně vytvořeno přímo v systému openMUC. V následujících kapitolách bude představeno, ale změny nejsou předpokládány. Zároveň UI souvisí silně s topologií celého systému. Protože se bude jednat o malý systém viz obr. 5.2, není nutné dělat nějaké rozsáhlé uživatelské prostředí. Z tohoto důvodu stačí již vytvořené Web UI pod systémem openMUC. Na začátku budou prvky IED virtualizovány pomocí již zmíněného programu ModBus Slave a posléze implementace přejde na protokol a přípravek IEC 68150. Hlavní zaměření bude na modře zvýrazněnou viz obr. 5.2 část topologie.



Obr. 5.2: Navržená topologie pro systém openMUC.

6 Realizace kybernetického prostředí openMUC

V následujících kapitolách bude realizován návrh zmíněný výše do nejvyšší možné míry. Bude zde například popsána prvotní inicializace. Způsob vkládání ovladačů a jejich správa skrze systém. Bude zde rozebráno webové prostředí openMUC a jak ho efektivně využívat nebo například jak přistupovat k jednotlivým kanálům navázaných na prostředí. Dále bude zanalyzováno navazovaní komunikace mezi koncentrátory a serverem, na kterém funguje openMUC.

6.1 Inicializace systému

Část inicializace sytému je velmi jednoduchá. Díky návrhu openMUC je možné spustit systém na různých operačních systémech. V příloze k diplomové práci je již zprovozněný systém openMUC, který stačí pouze inicializovat na chtěném počítači. Po stažení přílohy je potřeba nakopírovat složku ze systémem do cílové destinace, jenž si vybírá uživatel. Následně lze systém spustit dvěma způsoby. Pro systém Windows je spuštění realizováno přes .bat soubor, který je uložen ve složce bin hlavního adresáře. Pro systém Linux je ve stejném adresáři soubor „openmuc“. V terminálu je nezbytné navigovat do složky bin a poté spustit příkaz „./bin/openmuc start“. Pokud uživatel chce, může se rozhodnout spustit systém na popředí terminálu skrze parametr „-fg“.

Skript využívaný pro systém Linux je mnohem komplexnější než .bat soubor pro Windows, který je schopen pouze systém spustit. Linuxový skript má funkcionalitu navíc. Pomocí tohoto skriptu lze systém restartovat nebo nahrát konfiguraci bez restartu. Dále podporuje automatickou inicializaci systému při startu operačního systému, což je užitečné například při náhlém restartu HW, na němž openMUC funguje. V zásadě zde jde hlavně o následující příkazy vždy začínající „./bin/openmuc“ a pokračující příkazem:

- Stop – zastaví běh programu openMUC,
- Reload – nahraje novou konfiguraci z konfiguračního souboru,
- Restart – restartuje celou instanci.

V případě, že byl systém spuštěný na pozadí je možné k němu přistupovat pomocí „./bin/openmuc remote-shell“. Instanci lze vypnout pomocí standardních UNIXových zkratek. Po vykonání akce startu se nespustí žádné dialogové okno. Všechna funkcionalita systému se spravuje přes webové rozhraní, jehož představení se věnuje následující část.

6.2 Seznámení s WebUI

Za předpokladu, že inicializace systému proběhla bez problému (je potřeba sledovat výpisy z terminálu), openMUC instance je spuštěna. Uživatel na ni může přistoupit přes adresu Local Loopbacku na portu 8888. Za využití jakéhokoliv prohlížeče a po zadání 127.0.0.1:8888 se uživateli zobrazí následující okno viz obr. 6.1. Je potřeba zadat název účtu a heslo, což jsou ve výchozím stavu **admin/admin**. Jednotlivá okna byla zvýrazněna pro lepší orientaci.

Obr. 6.1: Hlavní okno správy openMUC.

Žlutě zvýrazněné aplikace nejsou pro tuto práci důležité. Jenom pro úplnost textu jsou zde zmíněny. V rámci Media Vieweru (4) je uložena základní dokumentace. Simple Demo Visualization (5) projekt byl vypnut, aby nezatěžoval systémové prostředky. Obsahuje demonstraci funkce openMUC, která však nebyla potřebná. Je zde pouze jako „placeholder“. Její výstup ve funkci je však možné vidět na obr. 4.6.

Nezvýrazněné dlaždice obsahují aplikace pro správu uživatele – User Configurator (6) a pro export dat z historie. V rámci aplikace pro správu uživatelů je možné přidat uživatele, nastavit heslo atd. Funkcionalita vhodná, pokud na systém přistupuje více lidí. V rámci této práce byl užíván pouze základní účet admin. Data Exporter (7) obsahuje aplikaci pro jednoduché vyexportování všech dat ze všech kanálů ve formátu CSV. Je zde možné si nastavit přesné rozpětí atd.

Zeleně zvýrazněné dlaždice jsou pro tuto práci nejdůležitější. Channel Configurator (1) je ta nejdůležitější aplikace, která je potřebná pro implementaci všech ovladačů a následné definování jednotlivých zařízení a kanálů. Channel Access Tool (2) je využíván pro správu kanálů. Je zde možné číst reálná data posílané koncovými zařízeními nebo jim udávat příkazy. Data Plotter (3) může být využit pro souhrnné zobrazování dat, pokud jsou dostupná reálná zařízení. Podporuje vyvolání historie dat, ale také živé čtení pro přehled o zařízení.

6.3 Inicializace ovladačů a zprovoznění topologie

Pro zprovoznění komunikace mezi openMUC a koncentrátory, případně simulovanými koncovými zařízeními, bylo potřeba zprovoznit a nahrát všechny potřebné ovladače. Ovladače byly zvoleny tak, aby co nejvíce reflektovaly zadání. Zároveň bylo potřeba, aby co nejvíce protokolů bylo podporováno ovladači v základu vzhledem k velké časové náročnosti implementovat jakýkoli ovladač. Dříve zmíněné taktéž velmi přispělo k volbě systému openMUC. Za použití openMUC instance přiložené v příloze této práce a při spuštění skriptu si je možné všimnout následujících řádků viz obr. 6.2. Zde lze pozorovat protokoly, které byly zvoleny pro implementaci. Za pomocí těchto protokolů je vždy navázána komunikace s koncovým prvkem. Každému z nich zde bude věnován prostor.

```
[OpenMUC Data Manager] INFO  o.o.f.core.datamanager.DataManager - Driver registered: dlms
[OpenMUC Data Manager] INFO  o.o.f.core.datamanager.DataManager - Driver registered: iec60870
[OpenMUC Data Manager] INFO  o.o.f.core.datamanager.DataManager - Driver registered: iec61850
[OpenMUC Data Manager] INFO  o.o.f.core.datamanager.DataManager - Driver registered: modbus
```

Obr. 6.2: Nahrávání ovladačů do jádra openMUC.

6.3.1 ModBus

ModBus jako „legacy“ protokol byl zvolen první pro realizaci. Je podporován valnou většinou reálných zařízení, takže v reálném nasazení by neměl být problém. Ovladač podporuje oba typy komunikace Modbusu. Což znamená, že jak způsob TCP, tak přes sériovou linku byl otestován. V rámci ovladače je podporováno i tzv. RTU přes TCP, kdy je RTU komunikace zapouzdřena a tunelována přes TCP. Tento způsob však nebyl testován, protože oba předchozí způsoby „nahrazují“ poslední zmíněný ve standardních a homogenních prostředích. K testování byl použit simulátor ModBus Slave (active)[34], který podporuje testování komunikace jak přes TCP, tak přes RTU.

Pokud se uživatel rozhodne využít sériového rozhraní, je naprosto nezbytné, aby byla správně nainstalována knihovna pro toto rozhraní **librxtx-java**. V případě, že nebude správně instalována, komunikace nebude fungovat. Pokud se uživatel pohybuje v systému na bázi Linux, je taktéž potřeba, aby byl přiřazen do skupin „dialout a plugdev“. Sériová linka má svá úskalí v konfiguraci, vzhledem k počtu parametrů. Pro přehlednost je níže uveden výpis z konfiguračního souboru „channels“ viz obr. 6.3.

Na obrázku si lze povšimnout nejdůležitějších parametrů. Parametr deviceAddess slouží k odkazovaní na požadované zařízení. Zde je uveden parametr „COM1“, protože virtuální zařízení bylo konfigurováno v systému Windows, pokud by se nacházelo v prostředí Linux bylo by zde například uvedeno „/dev/ttyS1“. Parametry vypsané v položce settings jsou nezbytné k fungování sériové linky. Nachází se zde modulační rychlost 9600 nebo zda jsou zde paritní bity atd.

```
<device id="modbud_testdeviceRTU">
  <deviceAddress>COM1</deviceAddress>
  <settings>RTU:SERIAL_ENCODING_RTU:9600:DATABITS_8:PARITY_NONE:STOPBITS_1:ECHO_FALSE:FLOWCONTROL_NONE:FLOWCONTROL_NONE</settings>
  <samplingTimeout>1s</samplingTimeout>
  <connectRetryInterval>5s</connectRetryInterval>
  <disabled>false</disabled>
</device>
```

Obr. 6.3: Konfigurace pro ModBus RTU.

Analogicky k sériové komunikaci je zde IP komunikace přes protokol TCP. Vzhledem k fungování dané komunikace je jednodušší nastavit požadované parametry. Pro srovnání je níže uveden výpis z konfiguračního souboru viz obr. 6.4. Avšak zde bylo naraženo poprvé na úskalí simulátorů. Je potřeba, aby instance openMUC a simulátoru fungovaly na rozdílných operačních systémech (mohou být stejného typu). Pokud tato podmínka není splněna, komunikace nefunguje a openMUC je ve stavu connecting, ale ne connected. V testovacím případě zde demonstrovaném, fungovala instance openMUC na virtualizovaném operačním systému Linux a simulátor na systému Windows. Výchozí port pro komunikaci ModBus TCP je 502, ale je možné ho změnit jak v openMUC, tak v simulátoru. IP adresa viditelná viz obr. 6.4 je adresa hostujícího systému, kde funguje simulátor. Ip adresu a port je vždy nezbytné zadat ve tvaru IP:PORT, aby bylo dosaženo komunikace. OpenMUC přijme dané nastavení i v jiných tvarech nebude však funkční. Parametr pro timeout je nastaven na 3000 ms, což je výchozí hodnota, tudíž pokládaná za nejlepší. Oproti tomu connectRetryInterval je zde nastaven na 1000 ms, což je kvůli rešení problémů v případě vypovězení konektivity. Je zde potřeba vzít v potaz, že ukazatel, zda je zařízení připojeno není spolehlivý na 100 % a je potřeba sledovat výpisy v terminálu.

```
<device id="modbus_testdeviceTCP">
  <deviceAddress>192.168.0.26:502</deviceAddress>
  <settings>TCP:timeout=3000</settings>
  <samplingTimeout>1s</samplingTimeout>
  <connectRetryInterval>1s</connectRetryInterval>
  <disabled>true</disabled>
</device>
```

Obr. 6.4: Konfigurace pro ModBus TCP.

Pokud se připojuje reálné zařízení, tak se nastaví přesně jako viz obr. 6.4. Vzhledem k tomu, že se nepředpokládá využití sériové linky v dnešní době, tak se využije pouze fungování ModBus TCP. Jakmile je dosaženo stavu connected, zpřístupní se funkce „scan for channels“. Tato funkce skenuje všechnu komunikaci mezi koncovým zařízením a openMUC instancí. Po oskenování vypíše všechny dostupné kanály a pomocí nich je již možné zaznamenávat reálná data nebo zadávat příkazy. Instance openMUC přiložená v příloze je na to připravena. Stačí pouze změnit IP adresy podle potřeby uživatele v zařízení „modbus_testdeviceTCP“.

6.3.2 IEC 60870-5-104

IEC 60870-5-104 je zástupce modernějšího přístupu ke SCADA systémům. Zároveň se však nejedená o nový protokol. Je značně využíván, a proto byl zvolen druhým protokolem. V předchozí části byl ModBus popsán podrobněji, protože do jisté míry je zde postup analogický. Akorát zde je využívána pouze IP architektura. Pro správné otestování komunikace zde byl použit simulátor ze serveru Sourceforge názvem IEC Server[35]. Simulátor byl zprovozněn na systému Windows a openMUC instance byla na virtualizovaném systému Linux. Pokud fungovaly obě instance na stejném systému, nedošlo ke správnému navázání komunikace. V případě, že instance fungovaly na oddělených systémech, tak byla navázána komunikace okamžitě a oba výpisy z Logu vypisovaly connected.

```
<device id="iec60870_testdevice">
  <deviceAddress>ca=1;port=2404;h=192.168.0.26</deviceAddress>
  <settings>mft=1000</settings>
  <samplingTimeout>1s</samplingTimeout>
  <connectRetryInterval>1s</connectRetryInterval>
  <disabled>true</disabled>
</device>
```

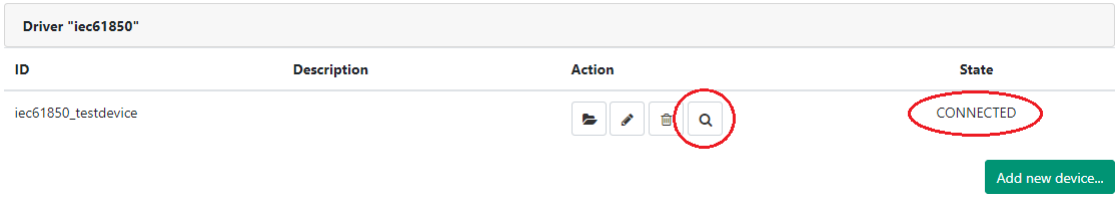
Obr. 6.5: Konfigurace pro IEC 60870-5-104.

Výše uvedená konfigurace viz obr. 6.5 je velmi podobná předcházející pro ModBus. Projevuje se zde přítomnost IP architektury, kdy je potřeba správně zadat IP a port. Je potřeba, aby byl uživatel opatrný na změnu syntaxe. Kdy port i IP se zadávají zvlášť jako parametry. Plus je zde parametr pro Common Address (ca=<?>) vycházející z fungování protokolu IEC 60870.

Ovladač podporuje čtení i zápis do valné většiny zařízení. Opět je zde funkce pro skenování kanálů. Za předpokladu využití reálného zařízení stačí pouze připojit, nahradit potřebné parametry (tzn. IP adresu, port, CA) a vše začne pracovat. Zmíněné 3 parametry jsou nezbytné, avšak ovladač podporuje specifikace různých nastavení. Například frekvenci opakování pokusů o připojení nebo timeout navazujících fragmentů (parametr mft).

6.3.3 IEC 61850

V rámci této části je věnován prostor IEC 61850. Ovladač je primárně zaměřený na komunikaci mezi koncentrátory a openMUC, což znamená na část standartu věnující se MMS. K ověření komunikace byl použit simulátor z diplomové práce Simulátor komunikace protokolů SCADA[21]. Jedná se o simulovanou verzi topologie zmíněnou v kapitole 3.3. Protokol opět využívá IP architektury.



Obr. 6.6: Výstup pro zařízení pracující pod IEC 61850.

Na výše uvedeném obrázku je výstup z openMUC prostředí. Pravé červené zvýraznění ukazuje na to, že je zařízení úspěšně připojeno. Z důvodu opakování je zde uvedeno pouze jednou. Stejné platí pro všechny protokoly. Komunikaci je vždy možné ověřit ještě v daném simulátoru nebo na reálném zařízení. Případně je možné filtrovat komunikaci a pozorovat „TCP 3-way handshake“ na jednotlivých paketech. Vhledem k tomu, že se jedná o standardní proces, je zde pouze zmíněn.

```
<driver id="iec61850">
  <samplingTimeout>0</samplingTimeout>
  <connectRetryInterval>1s</connectRetryInterval>
  <disabled>false</disabled>
  <device id="iec61850_testdevice">
    <deviceAddress>192.168.116.128:102</deviceAddress>
    <settings/>
    <connectRetryInterval>1s</connectRetryInterval>
    <disabled>false</disabled>
    <channel id="iec61850_testdevice_channel_66">
      <description/>
      <channelAddress>IEDPiRPi1/GGIO1.Health.stVal:ST</channelAddress>
      <unit/>
      <valueType>BYTE</valueType>
      <listening>true</listening>
    </channel>
    <channel id="iec61850_testdevice_channel_115">
      <description/>
      <channelAddress>IEDPiRPi1/GGIO1.SPCSO4.Oper.Test:CO</channelAddress>
      <unit/>
      <valueType>BOOLEAN</valueType>
      <listening>true</listening>
      <loggingInterval>1s</loggingInterval>
    </channel>
  </device>
</driver>
```

Obr. 6.7: Příklad konfigurace kanálu pod IEC61850.

Druhé zvýraznění ukazuje na funkci pro skenování kanálů, jež byla několikrát zmíněna. Simulátor použitý pro ověření IEC61850 podporuje tuto funkci. Pokud

je tedy využita, tak oskenuje celou komunikaci mezi koncentrátorem a openMUC. V tomto případě bylo nalezeno 242 kanálů. Příklad konfigurace se nachází na obrázku výše pro 2 z 242 kanálů viz obr. 6.7. Za pozornost stojí, že jakmile je instalováno všech 242 kanálů, tak dochází k silnému zpoždění reakcí celého systému. Jinak jsou parametry analogické k předchozím protokolům.

6.3.4 DLMS

DLMS je modernější protokol než předchozí. Poskytuje také vetší funkcionalitu. DLMS je primárně zaměřeno na komunikaci s chytrými měřícími přístroji. Podporuje komunikaci přes TCP/IP i protokol HDLC[36]. Bohužel v rámci diplomové práce nebyl nalezen simulátor věnující se tomuto prostředí a reálné zařízení nebylo k dispozici. Tudíž ovladač byl zařazen a zprovozněn, ale nebyla ověřena komunikace. Ovladač je připraven i na sériovou komunikaci. Konfigurace zařízení je však velmi podobná, až na chtěné parametry. V případě připojení zařízení lze však vybrat pouze, zda má uživatel zájem o využití TCP/IP nebo sériovou linku pomocí následujících parametrů oddělených vždy středníkem:

- t=serial nebo tcp,
- V závislosti na typu spojení pak stačí volit pro sériovou linku:
  sp=/dev/ttyS0 nebo COM0 ..... odpovídající operačnímu systému,
  bd=9600 ..... odpovídající konfiguraci linky,
- V závislosti na typu spojení pak stačí volit pro TCP/IP:
  h=192.168.0.26 ..... požadováná IP adresa,
  p=5081 ..... výchozí port pro DLMS,
- hdlc=true/false.

6.3.5 Komunikace přes privátní a veřejnou síť

Poslední kapitola se věnuje konektivitě a interoperabilitě systému. Stejně tak využití vizualizačních schopností openMUC. Vzhledem k nenáročnosti na systémové prostředky má openMUC možnost být instalován i na opravdu málo výkonné přístroje. V rámci této práce byl testován i na virtuálním stroji, který je ekvivalentní Raspberry Pi 3. Což znamená, že může být použit i v těch nejmenších aplikací. K těmto přístrojům je však potřeba přistupovat.

Pokud se uživatel pohybuje pouze v jedné směrovací doméně, tak je situace jednodušší. V privátní síti je možné si buď nastavit přímo IP adresu celého sytému v souboru framework/conf/systems.properties. V tomto souboru je nezbytné přidat parametr „org.apache.felix.http.host=<Host_Name/IP>“. Nebo je zde možnost přistupovat přímo na adresu systému, na kterém openMUC právě pracuje. Díky tomu, že funguje ve výchozím stavu na loopback adrese je možné se připojit pomocí

IP adresy hostitelského systému. Například v této práci to bylo velmi často na IP 192.168.0.26. Je zde však nutné definovat i port, který je možné změnit taktéž ve výše zmíněném souboru. Výchozí port, jak již bylo zmíněno, je 8888 pro protokol HTTP.

V eventualitě přistupování k serveru, který hostuje openMUC přes veřejnou síť jsou opět dvě možnosti. Každá má však nezbytnou predispozici. Je nezbytné vlastnit veřejnou IP adresu. První možnost je využít počítače/serveru v privátní síti za směrovačem do Internetu. Na daném směrovači je potřeba správně nastavit „port forwarding“. V případě testování v rámci této práce byla nastavena adresa 192.168.0.26 (adresa hostujícího serveru v rámci privátní sítě) a rozsah portů 8888-8889. Proč i 8889 bude vysvětleno dále. V případě, že je vše nastaveno v pořádku, tak pouze stačí zadat do prohlížeče uživatelovu veřejnou IP:port (například 62.XXX.XXX.XX8:8889). Rozhraní se posléze zobrazí.

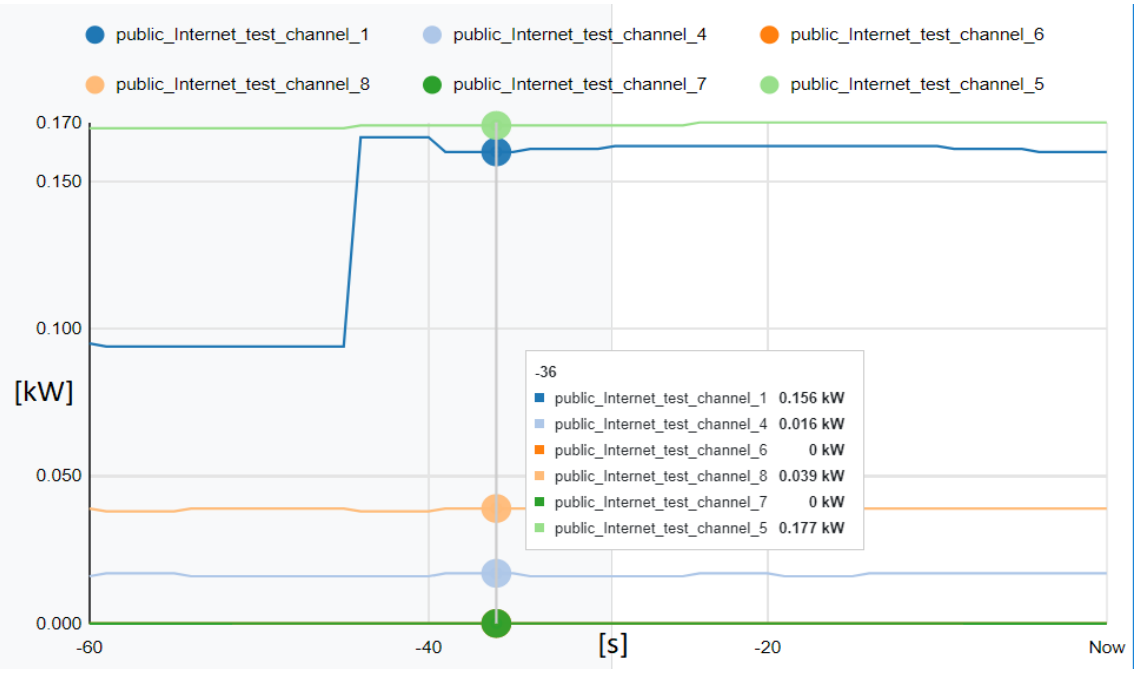
Nebo je zde alternativa k tomuto přístupu. V tomto případě je nutné, aby instance, na kterou uživatel chce přistupovat byla „hostována“ přímo na serveru s přidělenou veřejnou IP. Je zde možné využít ovladače pro REST server, který byl taktéž zprovozněn a otestován. Pomocí ovladače je možné číst všechny definované kanály na vzdáleném zařízení, aniž by se ně něj uživatel musel přímo připojit. Akorát je potřeba provést správné nastavení. Nastavení funguje, jakoby vzdálená openMUC instance byla další zařízení, které zabezpečuje REST server (v rámci openMUC prostředí). Takže je potřeba si nadefinovat zařízení pod ovladačem REST s následujícími parametry (je již připraveno, stačí změnit požadovanou IP):

- ID: např. public_Internet_test,
- Device Address:
  https://XXX.XXX.XXX.XXX:8889,
  http://XXX.XXX.XXX.XXX:8888,
- Settings: ct;admin:admin (přihlašovací údaje ke vzdálené instanci),
- Connect Retry Interval: např. 1000 ms.

Channel ID	Value	Time	Write
public_Internet_test_channel_1	0.152 kW	05/05/2019, 10:51:50	Write value Set record
public_Internet_test_channel_2	idle kW	05/05/2019, 10:51:50	Write value Set record
public_Internet_test_channel_3	15150 kW	05/05/2019, 10:51:50	Write value Set record
public_Internet_test_channel_6	0 kW	05/05/2019, 10:51:50	Write value Set record
public_Internet_test_channel_5	0.185 kW	05/05/2019, 10:51:50	Write value Set record
public_Internet_test_channel_4	0.016 kW	05/05/2019, 10:51:50	Write value Set record
public_Internet_test_channel_7	0 kW	05/05/2019, 10:51:50	Write value Set record
public_Internet_test_channel_8	0.039 kW	05/05/2019, 10:51:50	Write value Set record

Obr. 6.8: Výpis z rozhraní Channel Access Tool.

Posléze instance naváží konektivitu. Jako u ostatních ovladačů je možné využít funkce „Scan for channels“. Ta najde všechny aktivní kanály na daném spojení a vypíše je viz obr. 6.8. Zde je možné si povšimnout měnících se hodnot, které jsou aktualizovány každou vteřinu. Zde je funkce pro zapisování hodnot zpět do koncového zařízení. Čas poslední aktualizace atd. Všechny tyto hodnoty je možné vynést do grafu nebo exportovat do CSV souboru. V rámci samotného WebUI rozhraní je aplikace pro živý výpis do grafu. Její výstup je možné vidět níže viz obr. 6.9.



Obr. 6.9: Výpis z aplikace Data Plotter.

Pokud je komunikace vedena přes Internet (veřejnou síť), tak vyvstává otázka bezpečnosti. Bohužel dnes žádná komunikace nemůže zůstat nezašifrována. Hlavně pokud se jedná o „mission critical“ systémy, což obvykle valná většina SCADA systémů je. Zabezpečení pro openMUC se nachází na dvou vrstvách: transportní a aplikační. Výše je možné si všimnout využití dvou různých čísel portů. Jeden standardní 8888 a druhý 8889. Port 8889 je výchozí port pro zabezpečenou komunikaci pomocí protokolu HTTPS. Číslo portu lze opět změnit v souboru „systems.properties“, pokud uživateli nevyhovuje. Tudíž na aplikační vrstvě je využito protokolu HTTPS. V rámci transportní vrstvy bylo ověřeno pomocí programu Wireshark, že je využit protokol TLS pro navazování komunikace. Poslední forma obrany proti nechtěnému využití dat je samotný přístup do aplikace openMUC. V rámci této práce bylo ponechán výchozí účet admin/admin. Avšak tento účet, respektive jeho heslo je silně nedostatečné a slovníkový útok nebo i tzv. „bruteforce“ útok ho odhalí pod jednu vteřinu[37]. Důrazně je tedy doporučeno silné heslo.

Závěr

Práce na téma zadání Kybernetické prostředí pro systémy typu ICS/SCADA se skládá z 6 logických a navazujících celků. Jako první byl rozebrán Komunikační model systémů ICS/SCADA. V rámci této části byly definovány základní pojmy a elementární součásti příkladového SCADA systému. Prvky bylo nezbytné definovat, aby práce s těmito pojmy mohla pracovat a vyhodnocovat jejich roli ve výsledném systému. Teoretická část se skládá ještě z dalšího logického celku a to jsou nejvíce využívané protokoly. Bylo nezbytné pochopit, jak některé protokoly fungují a jakou mají funkcionalitu.

Praktická část práce se skládá ze 4 zbývajících celků. První je potřeba vybrat využívaný HW a SW. V této části jsou vyjmenovány a popsány všechny důležité komponenty a programy nezbytné pro úspěšné splnění zadání diplomové práce. Další část se zevrubně věnuje výběru SCADA systému. Tento výběr je velice podstatný a je potřeba mu věnovat dostatečný prostor, protože přímo ovlivňuje způsob řešení celé práce. Zde bylo důležité zjištění a také jedno z hlavních kritérií dostupnost dobré dokumentace plus podpory. K velkému překvapení mnohé ze systémů neobsahovaly dostatečnou dokumentaci. Podpora byla úplně neexistující u některých systémů. Takže nakonec výběr padl na 3 z 11 systémů. Zde už rozhoduje vybraný přístup k problematice. Všechny 3 systémy jsou dostatečně dobré. Pro tuto práci však byl zvolen openMUC.

V rámci návrhu kybernetického prostředí je demonstrován postup, kterým se práce nadále ubírá. Jsou představeny stavební kameny celého systému. Jsou zde nastíněny mantinely, jichž se práce drží. Následně práce plynule přechází k realizaci tohoto návrhu. Je zde rozebráno webové rozhraní. Jeho elementární využití a způsoby, jak s ním operovat. Jako prvopočátek na zprovoznění celého systému byl vybrán ModBus, vzhledem k jeho dostupnosti ve všech potřebných částech systému openMUC i koncových zařízení. Po úspěšné realizaci návrhu topologie v rámci protokolu MODBus bylo pokračováno s dalšími protokoly.

Hlavními byly zvoleny protokoly pod IEC. Oba byly úspěšně otestovány, avšak objevily se problémy z hlediska využitých simulátorů. Jako poslední byl implementován DLMS protokol, který však nemohl být řádně otestován, vhledem k nedostupnosti simulátoru. V rámci práce bylo podrobně popsáno, jak postupovat v konfiguraci jednotlivých aspektů všech různých protokolů. Na každém z protokolů byl demonstrována jedna z částí funkcionality openMUC systému. Na protokolu ModBus způsob připojení sériové linky. Na protokolu IEC 60870 způsob využití modernějšího přístupu TCP/IP. Na IEC 61850 skenování a využití kanálů. Na REST využití vizualizačních schopností openMUC atd.

Poslední část se věnovala způsobům připojení k instancím a propojení instancí openMUC. Kdy bylo dosaženo pomocí komunikace přes veřejnou síť i její zabezpečení pomocí protokolu HTTPS a TLS. Bylo zde vysvětleno, jakými způsoby lze propojit openMUC a jaká jsou úskalí jednotlivých možností. V rámci rozšíření práce by bylo možné uvažovat nad implementací dalších protokolů. Nabízí se zde DNP3, pro který je dostupná otevřená knihovna. Je však nezbytné celou knihovnu překlopit do formy ovladače pro openMUC, což si žádá velké prostředky. Dalším bodem by mohlo být připojení a nasazení do větších topologií se souběhem více způsobů komunikace pro ještě řádnější ověření robustnosti tohoto systému. V zasadě však bylo zjištěno, že stabilita je více než uspokojivá.

Literatura

[1] https://www.technickytydenik.cz/rubriky/ekonomika-byznys/od-1-prumyslove-revoluce-ke-4_31001.html

[2] STOUFFER, Keith; FALCO, Joe. Guide to supervisory control and data acquisition (SCADA) and industrial control systems security. National institute of standards and technology, 2006.

[3] CLARKE, Gordon; REYNDERS, Deon; WRIGHT, Edwin. Practical modern SCADA protocols: DNP3, 60870.5 and related systems. Newnes, 2004.

[4] DANEELS, Axel; SALTER, Wayne. What is SCADA?. 1999.

[5] GALLOWAY, Brendan; HANCKE, Gerhard P. Introduction to industrial control networks. IEEE Communications surveys & tutorials, 2012, 15.2: 860-880.

[6] Functional levels of a manufacturing control operation. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-12-5]. Dostupné z: https://en.wikipedia.org/w/index.php?oldid=893998573

[7] Kuwait elektrárna. Hpi-llc.com [online]. [cit. 2018-12-12]. Dostupné z: https://www.hpi-llc.com/expertise/maintenance-repair-overhaul-controls/turbine-controls/supervisory-control

[8] JUSOH, WNSE Wan, et al. Remote terminal unit (RTU) hardware design and implementation efficient in different application. In: Power Engineering and Optimization Conference (PEOCO), 2013 IEEE 7th International. IEEE, 2013. p. 570-573.

[9] RTU produkt. Scadaservis.cz [online]. [cit. 2018-12-12]. Dostupné z: http://www.scadaservis.cz/Produkty/Telemetrie/RTU/SCADAPack-500E

[10] BOLTON, W. Programmable logic controllers. 5th ed. Boston: Newnes, c2009. ISBN 978-1-85617-751-1.

[11] PLC produkt. Artelectro.ro [online]. [cit. 2018-12-12]. Dostupné z: http://artelectro.ro/nou/en/produse/automatizari-industriale/sisteme-de-automatizare/

[12] IEC 61158. Ec.europa.eu [online]. [cit. 2018-12-12]. Dostupné z: https://ec.europa.eu/eip/ageing/standards/ict-and-communication/interoperability/iec-61158_en

[13] HRUŠKA, František. Projektování řídících a informačních systémů. 2016.

[14] ARGHIRA, Nicoleta, et al. Modern SCADA philosophy in power system operation-A survey. University"Politehnica"of Bucharest Scientific Bulletin, Series C: Electrical Engineering, 2011, 73.2: 153-166.

[15] DUTERTRE, Bruno. Formal modeling and analysis of the Modbus protocol. In: International Conference on Critical Infrastructure Protection. Springer, Boston, MA, 2007. p. 189-204.

[16] KALAPATAPU, Rao. SCADA protocols and communication trends. ISA2004, 2004.

[17] MACKIEWICZ, Ralph E. Overview of IEC 61850 and Benefits. In: Power Systems Conference and Exposition, 2006. PSCE'06. 2006 IEEE PES. IEEE, 2006. p. 623-630.

[18] DEVARAJAN, Ganesh. Unraveling SCADA protocols: Using sulley fuzzer. In: Defon 15 Hacking Conf. 2007.

[19] Modbustools [online]. [cit. 2019-05-05]. Dostupné z: https://www.modbustools.com/modbus_slave.html

[20] Automatak [online]. [cit. 2019-05-05]. Dostupné z: https://www.automatak.com/opendnp3/docs/guide/current/#introduction

[21] STUDENÝ, Radim. Simulátor komunikace protokolů SCADA. Brno, 2017, 65 s. Semestrální projekt. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Petr Blažek

[22] Free-scada. Sourceforge.net [online]. [cit. 2019-04-23]. Dostupné z: https://sourceforge.net/p/free-scada/wiki/Home/

[23] Indigo SCADA. Enscada.com [online]. [cit. 2019-04-23]. Dostupné z: http://www.enscada.com/a7khg9/IndigoSCADA.html

[24] Opendax. Sourceforge.net [online]. [cit. 2019-04-26]. Dostupné z: https://sourceforge.net/projects/opendax/

[25] SEER2. Sourceforge.net [online]. [cit. 2019-04-26]. Dostupné z: https://sourceforge.net/projects/seer2/

[26] SCADA ProcessViewer. Sourceforge.net [online]. [cit. 2019-04-28]. Dostupné z: http://processviewer.sourceforge.net/

[27] ScadaBR. Scadabr.com [online]. [cit. 2019-04-28]. Dostupné z: http://www.scadabr.com.br/

[28] Szarp. Szarp.org [online]. [cit. 2019-05-05]. Dostupné z: https://szarp.org/en/

[29] MyScada. Myscada.org [online]. [cit. 2019-05-05]. Dostupné z: https://www.myscada.org/cs/

[30] Promotic. Promotic.eu [online]. [cit. 2019-05-05]. Dostupné z: https://www.promotic.eu/cz/index.htm

[31] OpenSCADA. Promotic.eu [online]. [cit. 2019-02-12]. Dostupné z: http://oscada.org/

[32] OpenMUC. Openmuc.org [online]. [cit. 2019-02-12]. Dostupné z: https://www.openmuc.org/

[33] OpenMUC Moduly. Openmuc.org [online]. [cit. 2019-04-30]. Dostupné z: https://www.openmuc.org/openmuc/

[34] ModBus Tool. Github.com [online]. [cit. 2019-04-30]. Dostupné z: https://github.com/graham22/modbustool

[35] IEC server. Sourceforge.net [online]. [cit. 2019-04-30]. Dostupné z: https://sourceforge.net/projects/iecserver/

[36] DE CRAEMER, Klaas; DECONINCK, Geert. Analysis of state-of-the-art smart metering communication standards. In: Proceedings of the 5th young researchers symposium. 2010. p. 1-6.

[37] Password-Cracking Times. Betterbuys.com [online]. [cit. 2019-05-05]. Dostupné z: https://www.betterbuys.com/estimating-password-cracking-times/

Seznam symbolů, veličin a zkratek

ICS	Industrial control system – Industriální systémy řízení
SCADA	Supervisory control and data acquisition – Dispečerské řízení a sběr dat
RTU	Remote terminal units – Vzdálení koncová jednotka
PLC	Programmable logic controller – Programovatelný logický automat
HMI	Human-machine interface – Rozhraní člověk-stroj
SW	Software
HW	Hardware
MTU	MTU – Master Terminal Unit
DCS	DCS – Data Control System
P2P	P2P – Point to Point
IEC	IEC – Intelligent Electronic Devices
IEC	IEC – International Electrotechnical Commission
DAQ	DAQ – Data acquisition

Seznam příloh

A Příloha

Diplomová práce obsahuje přílohu ve formě CD, na kterém se nachází zprovozněná instance openMUC programu. Všechny ovladače jsou v původním stavu vypnuty, aby si uživatel mohl zvolit protokol, jež zamýšlí použít ve své práci.