

System pro sběr a analýzu dat získaných z PLC

Mgr. Tereza Kovačiková

Bakalářská práce
2018



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2017/2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Mgr. Tereza Kovačková**
Osobní číslo: **A13117**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **kombinovaná**

Téma práce: **Systém pro sběr a analýzu dat získaných z PLC**
Téma anglicky: **A PLC Data Acquisition and Analysis System**

Zásady pro vypracování:

1. Provedte literární rešerši na téma Využití systémů pro sběr dat z programovatelných automatů.
 2. Popište způsob získávání dat z PLC Simatic firmy Siemens.
 3. Navrhněte databazový systém pro ukládání dat.
 4. Získaná data a sledované parametry vizualizujte pomocí libovolného grafického rozhraní.
-

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace, 1.díl. 1. Vyd. Praha: BEN technická literatura, 1999. ISBN 80-860-5658-9.
2. ŠMEJKAL, Ladislav. PLC a automatizace, 2.díl. 1. Vyd. Praha: BEN technická literatura, 2005. ISBN 80-7300-087-3.
3. MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty. Vyd. 2. Praha: Vydavatelství ČVUT, 2004. ISBN 80-010-2925-5.
4. SIMATIC Working with STEP 7 V5.5 (Getting Started) [online]. Norimberk: Siemens AG, 2010 [cit. 2017-01-30]. Dostupné z: <https://support.industry.siemens.com/cs/document/45531551/>
5. SIMATIC Programming with STEP 7 V5.5 (Programming Manual) [online]. Norimberk: Siemens AG, 2010 [cit. 2017-01-30]. Dostupné z: <https://support.industry.siemens.com/cs/document/45531107/>
6. SIMATIC NET Industrial Ethernet/PROFINET Industrial Ethernet (System Manual) [online]. Norimberk: Siemens AG, 2016 [cit. 2017-01-30]. Dostupné z: <https://support.industry.siemens.com/cs/document/27069465/>
7. BERGER, Von Hans. Automatisieren mit STEP 7 in AWL: speicherprogrammierbare Steuerungen SIMATIC S7-300/400. 2., wesentlich überarb. und erw. Aufl. Erlangen: Publicis-MCD-Verl, 1998. ISBN 38-957-8089-8.

Vedoucí bakalářské práce:

Ing. Tomáš Sysala, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

15. prosince 2017

Termín odevzdání bakalářské práce:

25. května 2018

Ve Zlině dne 15. prosince 2017

doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Jméno, příjmení: Tereza Kovačiková

Název bakalářské/diplomové práce: Systém pro sběr a analýzu dat získaných z PLC

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s tím, že licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považuji se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 23.5.2018


.....
podpis diplomanta

ABSTRAKT

Cílem práce je popsat způsob získávání dat z PLC SIMATIC firmy Siemens, následné uložení dat do databázového systému, odkud jsou dále předávána k vizualizaci, analýze a grafické prezentaci sledovaných parametrů. Je uveden všeobecný popis všech používaných prvků a komponent z kterých se celé řešení skládá, včetně PLC, sítě, databáze, ale i bezpečnost. Dále pak je uvedena konkrétní volba řešení včetně implementace jednotlivých kroků. Tato práce by měla poskytnout programátorům PLC řešení pro sledování systému v době jeho oživování, a také poskytnout pracovníkům údržby rychlou diagnostiku a možnost snížit doby prostojů během úprav nebo odstraňování problémů.

Klíčová slova: automatizace, vizualizace, sběr dat, PLC, SIMATIC, STEP 7, Snap7, Raspberry Pi, Python, Flask

ABSTRACT

The aim of the work is to describe the way of data acquisition from SIMATIC PLC by Siemens, the subsequent storage of data into the database system, from which they are further transmitted for visualization, analysis and graphical presentation of monitored parameters. A general description of all the elements and components used, including the PLC, the network, the database, as well as the security, are included. Here is a specific solution, including the implementation of individual steps. This work should be provided to PLC programmers system tracking at the time of the commissioning, and also provide diagnostics for the maintenance staff and the ability to reduce downtime while fixing or troubleshooting.

Keywords: automation, visualization, data acquisition, PLC, SIMATIC, STEP 7, Snap7, Raspberry Pi, Python, Flask

Chtěla bych poděkovat vedoucímu bakalářské práce panu Ing. Tomášovi Sysalovi, Ph.D., za cenné připomínky, rady a trpělivost. Dále bych chtěla poděkovat panu Ing. Vojtěchu Beranovi, DiS., za odborné konzultace. Poděkování patří také mé rodině za veškerou podporu během celého studia.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

OBSAH	7
ÚVOD	9
I. TEORETICKÁ ČÁST	10
1 PRŮMYSLOVÁ AUTOMATIZACE	11
2 PROGRAMOVATELNÉ AUTOMATY	12
2.1 FUNKČNÍ KOMPONENTY PLC	13
2.2 VNITŘNÍ ARCHITEKTURA	15
2.3 PROGRAMOVÁNÍ PLC	17
2.4 PLC SIMATIC	17
2.4.1 Základní vlastnosti PLC SIMATIC S7-300	18
2.4.2 Základní vlastnosti PLC SIMATIC S7-400	19
2.4.3 Základní vlastnosti PLC SIMATIC S7-1500	20
3 PRŮMYSLOVÉ POČÍTAČE	22
4 PRŮMYSLOVÉ SÍTĚ	24
4.1 TOPOLOGIE SÍTĚ	25
4.2 OSI MODEL	26
4.3 TCP/IP	27
4.4 ETHERNET	27
4.5 BEZDRÁTOVÁ KOMUNIKACE (Wi-Fi).....	28
5 VIZUALIZACE	30
5.1 SCADA	30
5.2 OPC	32
6 DATABÁZOVÉ SYSTÉMY	33
6.1 POSTGRESQL.....	33
6.1.1 Rozhraní pro správu databáze – pgAdmin	34
6.2 SQLITE	34
7 BEZPEČNOST V PRŮMYSLOVÉ AUTOMATIZACI	36
II. PRAKTICKÁ ČÁST	37
8 VOLBA PROGRAMOVACÍHO JAZYKA	38
8.1 VOLBA PROGRAMOVACÍHO JAZYKA	38
8.1.1 Python	39
9 KOMUNIKACE S PLC SIMATIC	40
9.1 KONFIGURACE PLC SIMATIC	40
9.2 STEP 7.....	42
9.3 SNAP7	42
9.3.1 Navázání spojení pomocí Snap7	43
9.3.2 Čtení a interpretace dat.....	49
9.4 HOSTITELSKÉ PC	51
9.4.1 Raspberry Pi 3 model B	51
10 NÁVRH DATABÁZOVÉHO SYSTÉMU	53

10.1	SQLITE	54
10.1.1	Instalace.....	54
10.1.2	Datové typy	54
10.2	POSTGRESQL.....	56
10.2.1	Datové typy	57
11	VIZUALIZACE ZÍSKANÝCH DAT	58
11.1	FLASK	58
11.2	KNIHOVNY PYTHONU	59
11.3	VIZUALIZOVANÁ DATA	59
	ZÁVĚR	66
	SEZNAM POUŽITÉ LITERATURY.....	67
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	71
	SEZNAM OBRÁZKŮ	72
	SEZNAM TABULEK.....	73
	SEZNAM PŘÍLOH.....	74

ÚVOD

S rostoucím množstvím dat nejen v oblasti průmyslové automatizace, ale i v různých oblastech lidské činnosti, vzrůstají požadavky na jejich efektivní pořízení, rychlé zpracování, srozumitelnou vizualizaci a vyhodnocení. Spolu s tím existuje také velké množství způsobů řešení.

Cílem práce je popsat způsob získávání dat z PLC SIMATIC firmy Siemens, následné uložení dat do databázového systému, odkud budou dále předávána k vizualizaci, analýze a grafické prezentaci sledovaných parametrů. Celé řešení má za cíl soustředit se na využití free a open-source produktů. Existuje celá řada komerčních produktů, které sběr dat z PLC umožňují, v této práci však bude uveden postup, jak pro tyto účely využít open-source knihovnu. Pro zaznamenaná data budou navrženy způsoby, jak je snadno uložit s důrazem na možnost využití databázových systémů, které je možno implementovat na operačních systémech Windows i GNU/Linux.

Pro interpretaci dat bude použito vhodné vizualizační rozhraní, které umožní parametry prohlížet jak z počítače, tak prostřednictvím mobilního zařízení. Získaná data budou znázorněna nejen v grafu, ale bude také k dispozici základní statistika sledovaných dat.

Cílem práce není pouze popsat principy sběru a zpracování dat, ale také poskytnout programátorům PLC návod k řešení, které by jim usnadnilo sledování systému v době jeho ožívování, a také následně pracovníkům údržby nabídnout rychlou a flexibilní diagnostiku, která je důležitá pro snižování dob prostojů během úprav nebo při odstraňování problémů za produkce i mimo ni.

Pro možnost dalšího rozšiřování aplikace budou uvedeny instalace jednotlivých programů a balíčků na operačních systémech Windows a GNU/Linux.

I. TEORETICKÁ ČÁST

1 PRŮMYSLOVÁ AUTOMATIZACE

Výrobní a procesní průmyslová automatizace se uplatňuje v řídicích procesech a informačních systémech.

V posledních desetiletích zaznamenal vývoj v oblasti automatizace velice rychlé tempo růstu a pokroku, což je dáno rozvojem v technologii, spolu s vyšším očekáváním uživatelů a pokročilosti průmyslových procesních technologií. [1]

Průmyslová automatizace je rozsáhlá a různorodá oblast zahrnující procesy, strojní zařízení, elektroniku, software a informační systémy, které spolupracující a jejichž kooperace vede k dosažení cílů v oblasti výroby, vylepšení kvality, nižší ceně a maximální flexibilitě. [10]

2 PROGRAMOVATELNÉ AUTOMATY

Programovatelný automat je uživatelsky programovatelný řídicí systém přizpůsobený pro řízení průmyslových a technologických procesů. Nejčastěji se označuje zkratkou PLC (Programmable Logic Controller). Jde o průmyslový počítač používaný pro automatizaci procesů v reálném čase – řízení strojů, nebo výrobních linek v průmyslovém prostředí. [1]

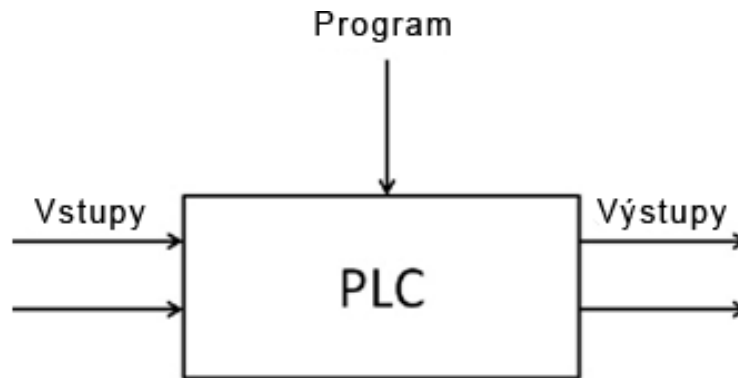
V roce 1970 byl představen první programovatelný automat jako odpověď na požadavek General Motors na polovodičový systém, který by měl vlastnosti počítače a navíc by mohl být programován a spravován inženýry a techniky v továrně.

První programovatelné automaty byly méně prostorově náročné a nahradily relé, čítače, časovače a jiné kontrolní komponenty, zároveň nabídly více flexibility co se týče jejich možnosti přeprogramování a zároveň nabídly lepší možnosti nastavení.

První programovací jazyk byl založen na žebříkovém diagramu a na elektrických symbolech běžně používaných elektrikáři, což bylo klíčové k přijetí programovatelného automatu v průmyslu.

Jelikož může být programovatelný automat programován v žebříkové logice, bylo relativně jednoduché převést elektrické schéma do programu programovatelného automatu. Tento proces zahrnuje definování operačních pravidel pro každý kontrolní bod, převedení těchto pravidel do žebříkové logiky a identifikaci a označení výstupů (adresace).

Programovatelný automat (PLC) je považován za hlavní stavební jednotku automatizačních systémů. PLC je speciální forma mikroprocesorového kontroléru, který používá programovatelnou paměť k ukládání instrukcí a implementaci logických, sekvenčních, časových, číselných a aritmetických funkcí k řízení strojů a procesů, tak jak je vyobrazeno (Obr. 1), a je navržen tak, aby mohl být obsluhován techniky s omezenou znalostí počítačů a počítačových jazyků. [10]



Obr. 1. Schéma PLC. [10]

PLC jsou podobné počítačům, ale počítače jsou optimalizovány pro výpočetní a zobrazovací úkony, PLC jsou optimalizovány pro řídicí úkony a průmyslové prostředí. PLC jsou:

- Robustní a odolné vůči vibracím, teplotě, vlhkosti a hluku.
- Vybaveny rozhraními pro vstupy a výstupy.
- Snadno programovatelné a používají srozumitelný programovací jazyk, který se primárně zaměřuje na logické operace a spínání.

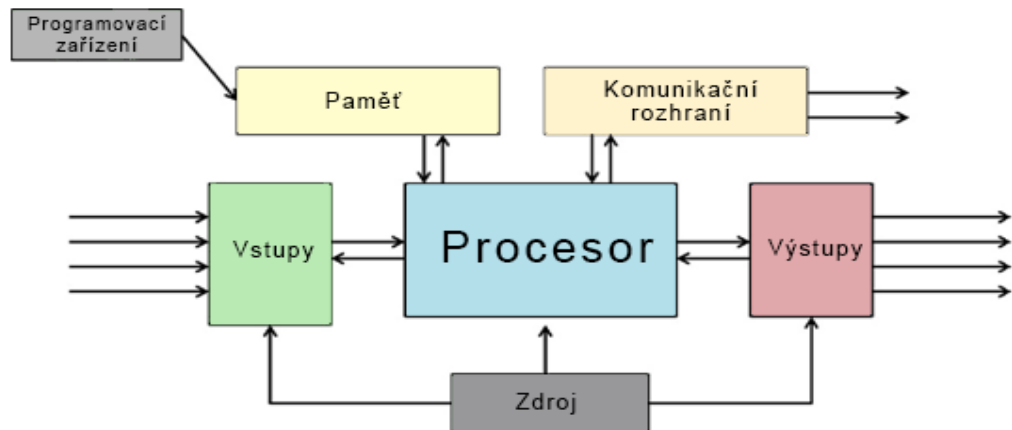
V dnešní době jsou PLC široce používány v podobě malých samostatných jednotek až po modulární systémy s velkým množstvím vstupů a výstupů, digitálních i analogových, a také vykonávají proporcionálně-integračně-derivační řídicí procesy (PID).

2.1 Funkční komponenty PLC

Typický PLC systém obsahuje základní funkční komponenty jako je procesorová jednotka, paměť, zdroj elektrické energie, vstupní/výstupní (I/O) rozhraní, komunikační rozhraní a programovací zařízení (Obr. 2).

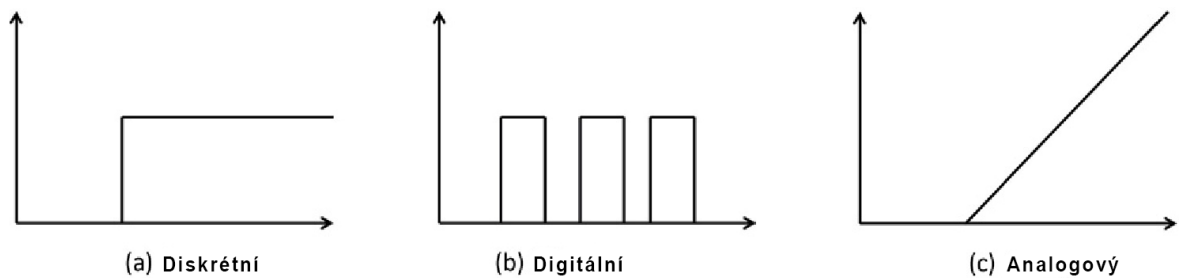
- Procesorová jednotka a centrální procesorová jednotka (CPU) obsahují mikroprocesor, který překládá vstupní signály a zajišťuje řídicí akce na základě programu uloženého v paměti a výsledky posílá jako signály na výstup.
- Zdroj elektrické energie je potřeba k převedení střídavého síťového napětí na stejnosměrné napětí nezbytné pro procesor a obvody ve vstupních a výstupních rozhraních modulů.

- Programovací zařízení (PG, něm. das Programmiergerät), nebo osobní počítače (PC) s programovacím softwarem jsou používány k nahrání programu do paměti procesoru. Program je napsán v zařízení (PC/PG) a poté přesunut do paměťové jednotky v PLC. [10]



Obr. 2. Základní uspořádání PLC. [10]

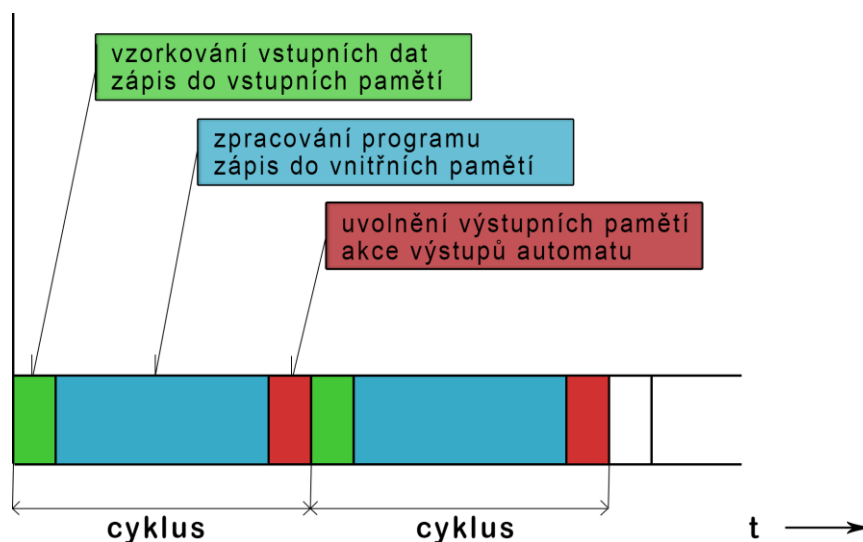
- Paměťová jednotka slouží k uložení programu, který obsahuje instrukce prováděné mikroprocesorem, data uložená ze vstupu určená ke zpracování a data určená pro výstup.
- Vstupní a výstupní rozhraní jsou místa, kde procesor přijímá informace z externích zařízení a předává informace externím zařízením. Vstupy tak mohou být ze spínačů nebo ze senzorů jako jsou fotoelektrické články, také čítací zařízení, teplotní senzory, nebo průtokové senzory. Výstupy mohou být například startéry motorů, solenoidové ventily atd.
- Vstupní a výstupní zařízení mohou zpracovávat signály, které jsou diskrétního, digitálního nebo analogového charakteru (Obr. 3). Signály zařízení, které poskytují diskrétní nebo digitální signály, nabývají také hodnot vypnuto (OFF) nebo zapnuto (ON).



Obr. 3. Diskrétní, digitální a analogový signál. [10]

- Komunikační rozhraní je používáno pro přijímání a vysílání dat prostřednictvím komunikační sítě z nebo do jiných vzdálených PLC. Má na starosti takové akce, jako je potvrzování, sběr dat a synchronizaci mezi uživatelskými aplikacemi. [10]

PLC umožňuje režim reálného času a hraje významnou roli v konkurenceschopnosti PLC oproti IPC a dalším prostředkům průmyslové automatizace. Způsob práce, který od počátku charakterizuje PLC a odlišuje je od řídicích mikropočítačů, tj. cyklické vykonávání programu. Ten zůstal základním režimem prakticky všech PLC (Obr. 4).

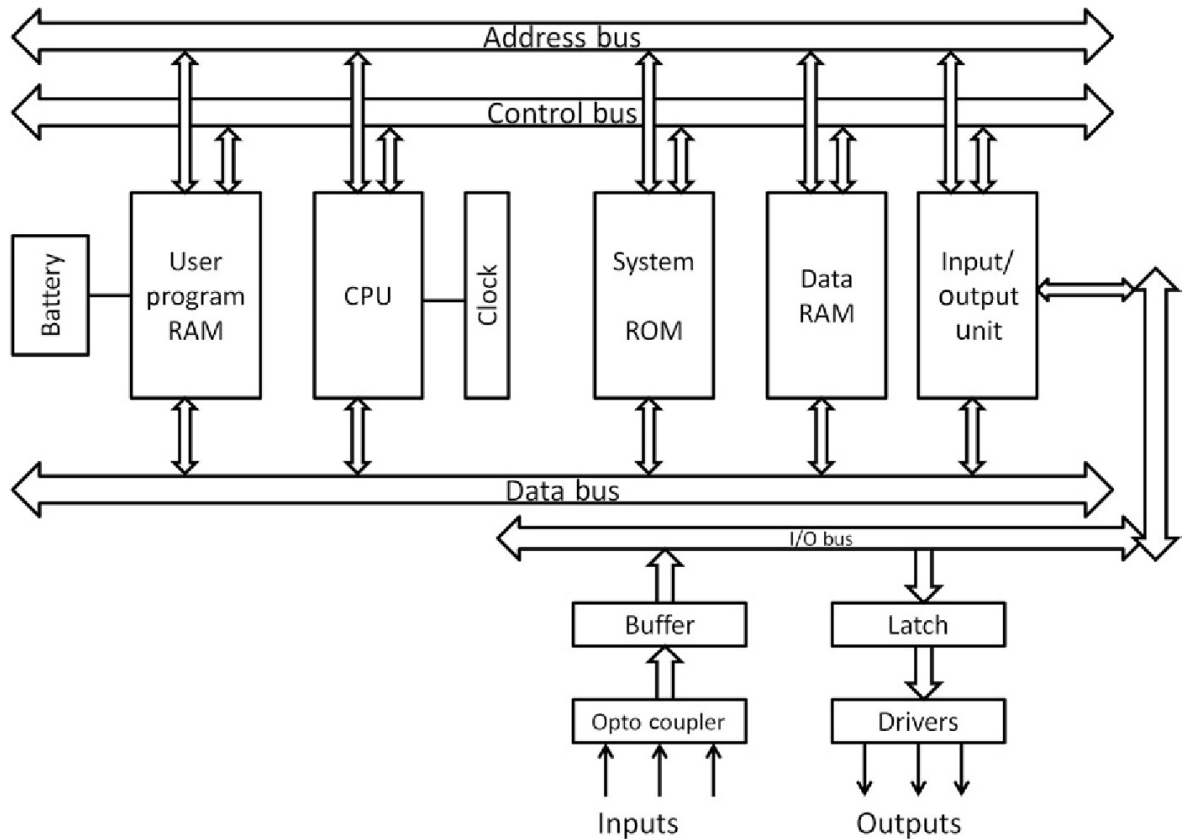


Obr. 4. Cyklické vykonávání programu. [15]

2.2 Vnitřní architektura

Základní vnitřní uspořádání PLC je na obrázku (Obr. 4). Skládá se z CPU obsahující systémový mikroprocesor, paměť a vstupní a výstupní obvody. CPU řídí a zpracovává

všechny operace uvnitř PLC. Je vybaven časovačem s frekvencí běžně mezi 1 a 8 MHz. Tato frekvence rozhoduje o rychlosti PLC a poskytuje časování a synchronizaci pro všechny prvky v systému.



Obr. 5. Vnitřní uspořádání PLC. [10]

Informace v PLC jsou přenášeny v podobě digitálních signálů. Vnitřní dráhy digitálních signálů se nazývají sběrnice. Po fyzické stránce je sběrnice jen seskupení vodičů s elektrickým signálem. Mohou být v podobě tištěných spojů nebo plochých vícežilových kabelů. CPU využívá datové sběrnice k posílání dat mezi jednotlivými prvky, adresovou sběrnici k lokaci pro přístup k uloženým datům a řídicí sběrnici pro signály, které souvisí s interním řízením komunikace.

Systémová sběrnice je používána pro komunikaci mezi vstupními a výstupními porty a vstupní a výstupní jednotkou. [10]

2.3 Programování PLC

PLC mohou být programována pomocí mobilních zařízení, desktopové konzole nebo počítače. Program je navržen na programovacím zařízení a přenesen do paměťové jednotky PLC.

U některých PLC je vyžadován počítač s příslušným software; jiné vyžadují speciální komunikační karty k propojení s PLC. Hlavní výhodou použití počítače je to, že program může být uložen na harddisku nebo na CD a také může být sdílen na síti či kopírován. [10]

U PLC převažují tyto programovací jazyky:

A. Grafické

Jazyk reléových schémat – Ladder Diagram (LD)

Jazyk logických schémat – Function Block Diagram (FBD)

B. Algebraické

Memo kód (SL –Statement List)

Normalizaci HW a zejména způsob programování PLC udává standard IEC 1131-3. [15]

Datové typy a proměnné

Definice datových typů jsou obdobné jako u vyšších programovacích jazyků. Standard pracuje s předdefinovanými typy BOOL, INT, WORD, DWORD a umožňuje odvozování dalších uživatelských typů včetně polí a vícerozměrných polí.

Základní vlastnosti deklarací a definicí podle IEC 1131-3 jsou tedy v podstatě totožné s vlastnostmi deklarací a definicí ve vyšších programovacích jazycích. Standard je však dále rozšiřuje pro speciální potřeby programovatelných automatů. [15]

2.4 PLC SIMATIC

V této práci bylo realizováno spojení s modely z řad SIMATIC S7-300, S7-400, S7-1500, dále je uvedena jejich podrobnější specifikace. Přehled modelů, které se v současnosti používají:

V současnosti jsou na trhu dostupné řady PLC SIMATIC: S7-200, S7-300, S7-400, S7-1200, S7-1500. Předchůdci stávající řady jsou PLC SIMATIC S5

2.4.1 Základní vlastnosti PLC SIMATIC S7-300

- Modulární mini PLC pro jednodušší a středně složité aplikace.
- Spektrum výkonově rozdílných CPU jednotek.
- Rozsáhlé spektrum rozšiřujících periferních modulů.
- Rozšiřitelnost přístroje až do konfigurace s 32 moduly.
- Systémová sběrnice integrovaná uvnitř jednotek a modulů.
- Možnost připojení do průmyslové sítě:
 - MPI (Multipoint Interface)
 - PROFIBUS
 - Industrial Ethernet
- Jednotné rozhraní pro připojení PG/PC přístroje s přístupem ke všem modulům konfigurace PLC přístroje
- Žádná omezení pro kombinaci jednotek a periferních modulů
- Parametrizace sestavy a nastavování parametrů pomocí programu – nástroje „HWConfig“ ze softwarového nástroje STEP 7.
- Komunikační procesory (CP) – slouží výhradně pro připojení PLC přístroje do některého z dostupných síťových datových systémů:
 - Point-to-point spojení
 - PROFIBUS
 - Industrial Ethernet a PROFINET [41]

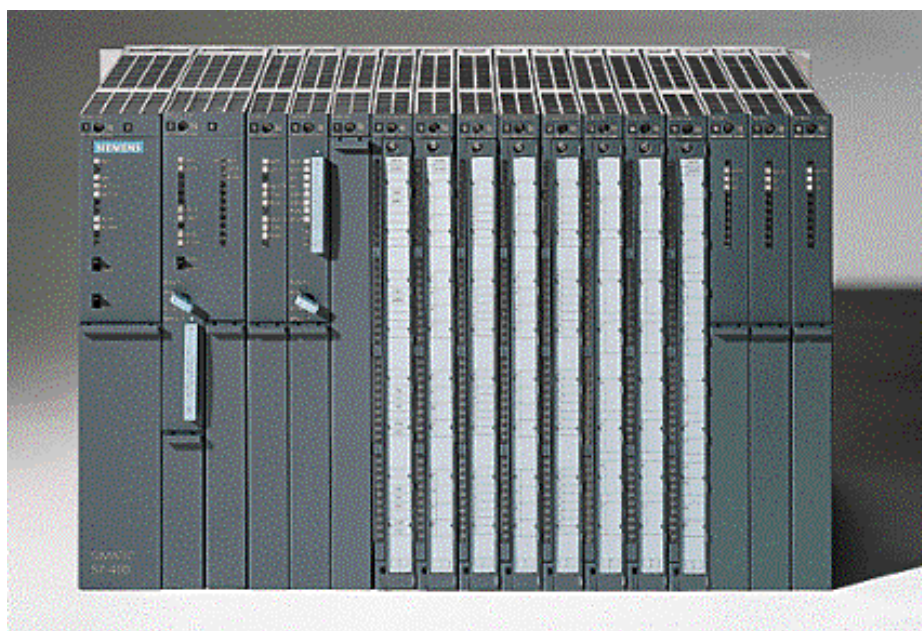


Obr. 6. PLC SIMATIC S7-300, CPU. [41]

2.4.2 Základní vlastnosti PLC SIMATIC S7-400

- Výkonný modulární PLC přístroj určený pro střední až nejnáročnější automatizační aplikace
- Řada výkonově rozdílných CPU jednotek
- Rozsáhlé spektrum rozšiřujících periferních modulů
- Rozšiřitelnost přístroje až do sestavy o 300 modulech
- Systémová sběrnice integrovaná do nosiče modulů:
 - Možnost připojení do průmyslové sítě: MPI (Multipoint Interface)
 - PROFIBUS
 - Industrial Ethernet
- Jednotné rozhraní pro PG/PC přístroj umožňující komunikaci se všemi moduly v konfiguraci
- Žádné omezení pro kombinaci jednotek a periferních modulů
- Parametrizace sestavy a nastavování parametrů pomocí programu-nástroje HWConfig ze softwarového nástroje STEP 7

- Víceprocesorový režim činnosti (až 4 CPU jednotky mohou být osazeny do jednoho racku)
- Komunikační procesory (CP) – slouží výhradně pro připojení PLC přístroje do některého z dostupných síťových datových systémů:
 - Point-to-point spojení
 - PROFIBUS
 - Industrial Ethernet a PROFINET



Obr. 7. PLC SIMATIC S7-300, CPU. [42]

2.4.3 Základní vlastnosti PLC SIMATIC S7-1500

PLC SIMATIC S7-1500 patří k nejnovějším modelům společnosti Siemens.

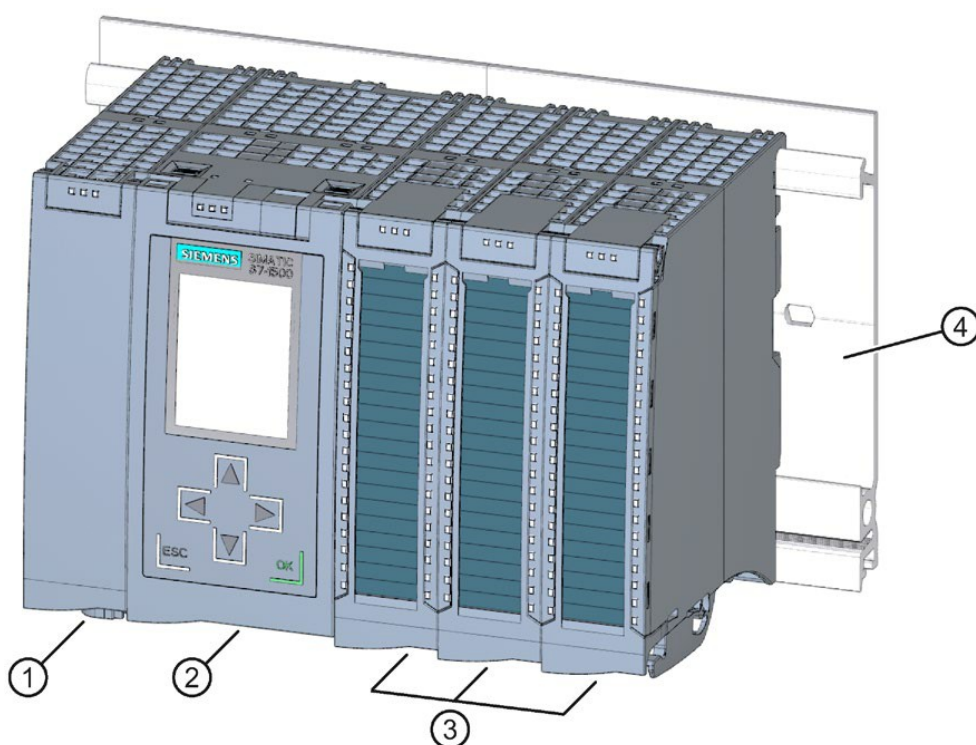
Automatizační systém SIMATIC S7-1500 se skládá z následujících komponent:

- CPU (standard, fail-safe, kompaktní nebo technologická CPU)
- Digitální a analogové vstupní a výstupní moduly
- Komunikační moduly (PROFINET/Ethernet, PROFIBUS, point-to-point)
- Napájecí systém (volitelný)

S7-1500 se instaluje na montážní lištu. Skládá se až z 32 modulů (CPU, napájecí systém a 30 vstupních a výstupních modulů). [42]



Obr. 8. PLC SIMATIC S7-1500. [39]



Obr. 9. Příklad konfigurace automatizačního systému S7-1500. [39]

1 – Napájecí modul 2 – CPU, 3 – Vstupní a výstupní moduly, 4 – Montážní lišta s integrovanou DIN lištou

3 PRŮMYSLOVÉ POČÍTAČE

Jednou z moderních variant řídicích systémů je osobní počítač PC v průmyslovém provedení nazývaný IPC (Industry PC).

IPC dodává průmyslovému řízení podobný programátorský komfort, jaký poskytovaly řídicí počítače a minipočítače a jaké ve velké míře poskytují distribuované systémy pro řízení procesů (DCS). To však splňuje každé PC. Avšak pro řízení je nezbytně nutné, aby řídicí systém splňoval ještě celou řadu elektrotechnických norem a doporučení.

Od poloviny 90. let nabývá podíl IPC na trhu automatizace na významu a je jasné, že v jisté době budou vážným konkurentem PLC a další řídicí technice. Na vyšší úrovni řízení vytlačují výkonná PC drahé pracovní stanice, avšak i na úrovni bezprostředního řízení roste jejich význam. Již dnes je varianta řídicího systému na bázi IPC levnější než ekvivalent v PLC provedení. Stále však PLC vykazují vyšší spolehlivost a tím i bezpečnost, což jsou dva dnes nejsledovanější parametry řízení. Důvodem je stále vysoká spolehlivost jednoduchého, multitaskingového operačního systému reálného času, kterým disponují PLC. Ani Windows NT¹, které jsou na IPC používány, nemohou být těmito systémům rovnocenným konkurentem ve stabilitě. Vývoj ovšem hovoří pro IPC a podíl PLC na řízení (zejména ve větších aplikacích, kde nedostatečný programátorský komfort prodlužuje fázi projektování řídicího systému) bude klesat. V současné době řeší IPC úlohy řízení logické úrovně, přesněji úkoly bezprostředního řízení a regulace s převažujícím logickým řízením. [15]

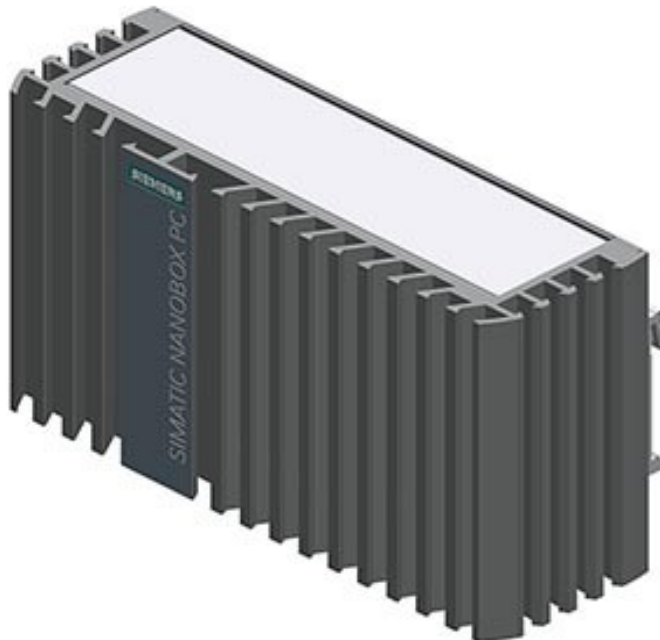
Kompaktní automat, který kombinuje prvky a schopnosti PC a PLC se označuje jako PAC (Programmable Automation Controller).

- PLC
 - Modulární uspořádání, spolehlivost.
 - Široké spektrum I/O modulů a konfigurací systému.

¹Do rodiny Windows NT patří: Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows NT 4.0, Windows 2000, Windows XP, Windows Server 2003, Windows XP (x64), Windows Server 2003 R2, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Phone 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10TP, Windows 10, Windows Server 2016

- PC
 - Velká paměť a vysokorychlostní zpracování.
 - Vysokourovňové zpracování dat.
 - Široké možnosti zasíťování, široká škála protokolů.

PAC je často používán v průmyslu pro řízení procesů (process control), sběr dat, vzdálený monitoring zařízení, dohled nad stroji a řízení pohybu.



Obr. 10. Příklad PAC – SIMATIC IPC227E (Nanobox PC). [34]

PAC splňují komplexní požadavky moderních aplikací průmyslové automatizace, jelikož kombinují prvky tradičních automatizačních technologií jako je PLC distribuovaných řídicích systémů (DCS), terminálů pro vzdálenou správu (RTU) a osobních počítačů (PC).

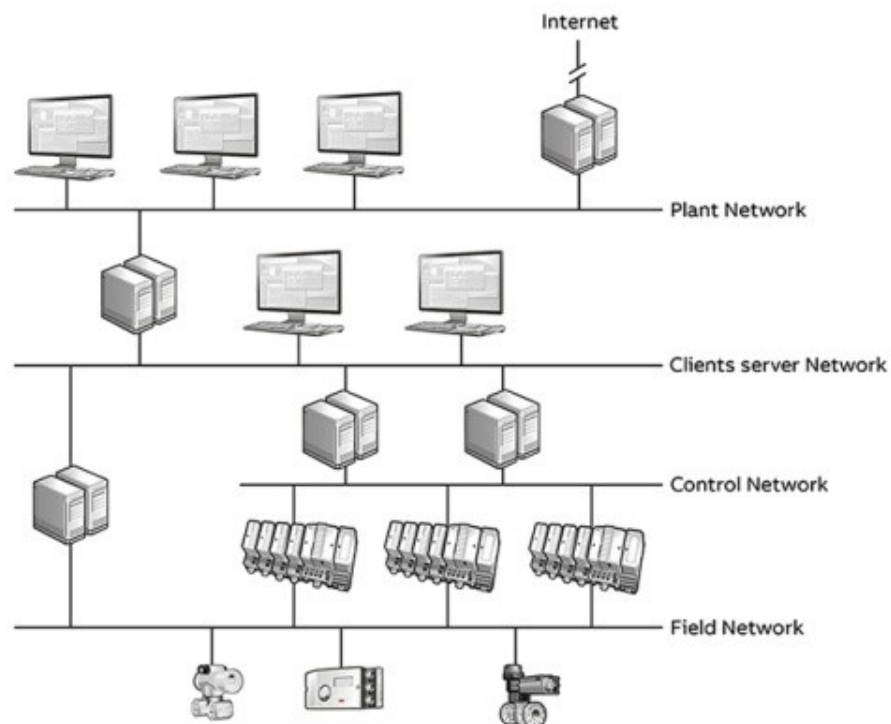
V průmyslu mohou být používány také jednočipové počítače, jak uvádějí např. práce autorů Razy a kol., Molana nebo Winna. Jejich nasazení někdy souvisí s požadavky na nízké pořizovací náklady. [18, s. 209; 19, s. 521; 20]

4 PRŮMYSLOVÉ SÍTĚ

Dnešní automatizace není jen jedno řídicí zařízení, které něco řídí. V současné době je trend sdružování řídicích komponent do sítí, kde jednotlivá zařízení komunikují mezi sebou nebo s nadřazeným systémem. A dále také vybavení řídicích jednotek komunikačními kanály, pomocí kterých se přes síť můžeme připojit do zařízení odkudkoliv na světě.

Nejdůležitější vrstvy průmyslové sítě jsou nazývány:

- Field Network, někdy také tzv. „fieldbus“
- Plant control network (PCN)
- Plant information network (PIN)



Obr. 11. Vrstvy průmyslové sítě. [22]

Sítí je označován počítačový hardware, software, kabely, síťová zařízení jako jsou přepínače, směrovače a ostatní zařízení, které počítačům dovolují vzájemnou komunikaci.

[10]

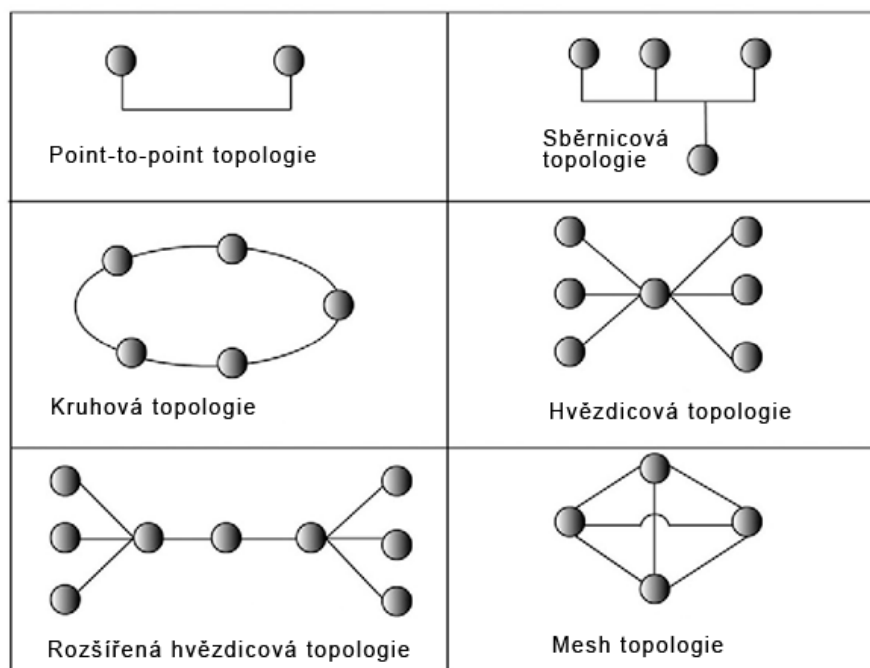
4.1 Topologie sítě

Topologie popisuje schéma propojení různých počítačových prvků. Jsou dva typy síťových topologií, jmenovitě fyzická a logická. Fyzická topologie znázorňuje rozvržení síťových kabelů, jejich propojení s počítači a jinými síťovými zařízeními. Logická topologie definuje, jak jsou data přenášena z jednoho zařízení na jiné bez ohledu na propojení hardwaru v síti.

Logická topologie je obecně definována síťovým protokolem, zatímco fyzická topologie je definována hlavně fyzickým rozložením síťových kabelů a zařízení.

Níže je uvedeno několik fyzických síťových topologií:

- Point-to-point topologie
- Sběrníková topologie
- Kruhová topologie
- Hvězdicová topologie
- Rozšířená hvězdicová topologie
- Mesh topologie



Obr. 12. Síťová topologie. [10]

Dříve dodavatelé vyvíjeli ke komunikaci mezi jejich vlastními produkty utvářejícími proprietární sít' svoje vlastní standardy a protokoly. Ale s růstem použití počítačů a rostoucí poptávkou po sdílení zdrojů v síti se nakonec počítačový svět posouvá od modelu proprietárních technologií k otevřeným standardům a protokolům. Otevřené standardy nejsou vlastněny žádnou společností nebo výrobcem, ale jsou spravovány standardizačními komisemi, jako jsou například Organization for Standardization, Institute of Electrical and Electronic Engineers (IEEE) a další. [10]

4.2 OSI model

Úlohou referenčního modelu OSI je poskytnout společnou základnu pro koordinované vypracování norem pro účely snadného a funkceschopného propojení tzv. otevřených systémů. Referenční model se týká výměny informace mezi otevřenými systémy (týká se pouze síťového vybavení), které nejsou specifikovány nějakou implementací, technologií nebo určitým způsobem propojení, ale tím, že podporují příslušné normy. [10]

Každé zařízení, které tvoří samostatný celek schopný vykonávat zpracování a přenos informace (počítač, terminál, periferní zařízení, apod.), je v terminologii OSI reálný systém. Pokud je jeho síťové vybavení v souladu s OSI, hovoří se o reálném otevřeném systému. [13]

Tab. 1. Referenční model OSI a jednotlivé funkce vrstev.

Název	Funkce
Vrstva 7: Aplikační	Rozhraní s uživatelem – procesy aplikací
Vrstva 6: Prezentační	Reprezentace dat
Vrstva 5: Relační	Komunikace-relace mezi stanicemi
Vrstva 4: Transportní	Přenos dat mezi koncovými uzly
Vrstva 3: Síťová	Adresace a směrování
Vrstva 2: Linková	Přístup k přenosovému prostředku
Vrstva 1: Fyzická	Přenos bitů

4.3 TCP/IP

TCP/IP byl vyvinutý ministerstvem obrany USA a je nazýván TCP/IP model nebo DoD model. Jedná se o otevřený standard. TCP/IP (Transmission Control Protocol/Internet Protocol) označuje celou síťovou architekturu. Na rozdíl od sedmivrstvého referenčního modelu OSI protokolovou architekturu TCP/IP tvoří jen čtyři vrstvy (tabulka č. 1)

Tab. 2. Porovnání architektury TCP/IP s referenčním modelem OSI. [13]

OSI	TCP/IP
Aplikační vrstva	Aplikační
Prezentační vrstva	
Relační vrstva	
Transportní vrstva	Transportní
Síťová vrstva	Vrstva Internetu
Spojová vrstva	Síťové rozhraní
Fyzická vrstva	

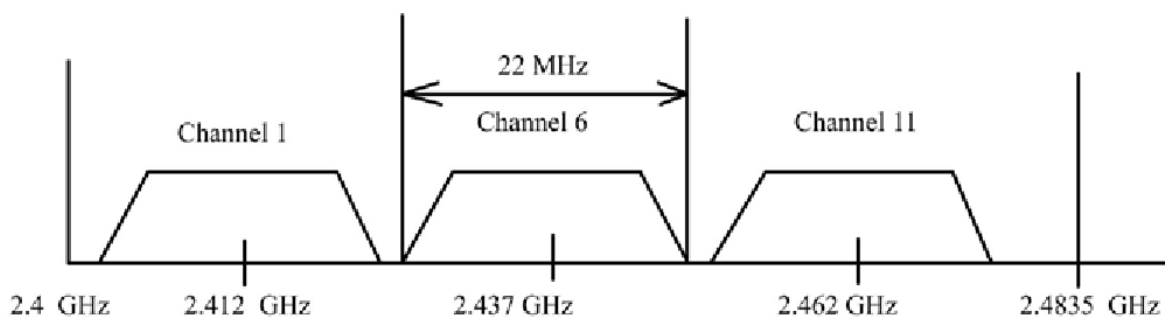
4.4 Ethernet

Ethernet je sériová sběrnice vyvinutá na konci 70. let firmou Xerox, později byl Ethernet standardizován jako norma IEEE 802.3. Vychází ze specifikace IEEE 802.3 pro fyzickou vrstvu a doplňuje ji o specifikaci linkové vrstvy, zejména horní podvrstvy definující LLC (Logical Link Control – řízení logického spoje). Dolní podvrstva linkové vrstvy, MAC (Medium Access Control), definující způsob přístupu k přenosovému médiu, je charakterizována jak u Ethernetu, tak u IEEE 802.3 nedeterministickou přístupovou metodou CSMA/CD (Carrier Sense Multiple Access/Collision Detection). [15] V průmyslové automatizaci se dále používají protokoly ModBus, PROFINET a průmyslový Ethernet. Tato práce je realizována přes Ethernet, nebo průmyslový Ethernet a Wi-Fi.

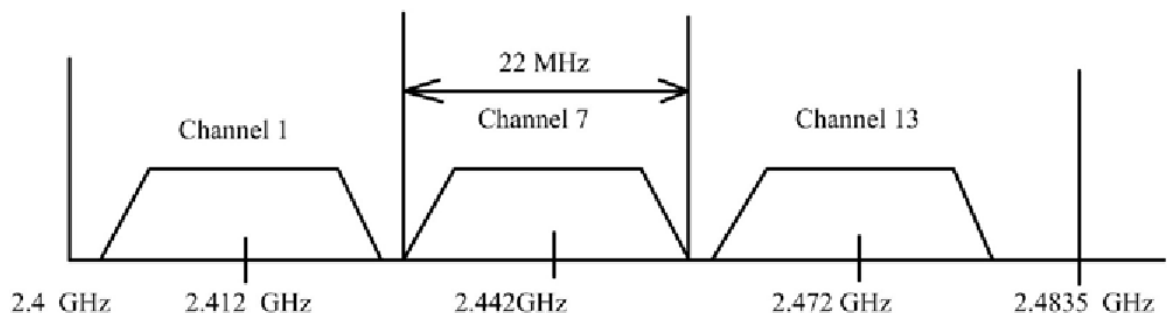
4.5 Bezdrátová komunikace (Wi-Fi)

V mnohých průmyslových a výrobních podnicích se již běžně pracuje s bezdrátovými zařízeními. Existuje mnoho výhod, které bezdrátová zařízení mají, například v souvislosti se sběrem dat a řízením systémů, které tlačí na přechod na bezdrátovou přístrojovou techniku. Jakákoliv bezdrátová komunikace nebo přenos dat se primárně odehrává mezi dvěma zařízeními nazývanými vysílač (transmitter) a přijímač (receiver).

Bezdrátová komunikace závisí na několika faktorech. Ale hlavním faktorem je frekvenční spektrum. Některá pásma jsou pro komerční účely licencovaná, což stojí nemalé částky. Existuje ale také několik frekvenčních rozsahů, které nejsou licencovány. Do této skupiny patří také průmyslová bezdrátová komunikace, kde se toto pásmo označuje jako ISM (industrial, scientific and medical) s frekvencí 2,4 GHz. Jedná se o rozsah 2402–2482 MHz (v některých zemích do 2483,5 MHz). [10]



(a) IEEE 802.11b North American Channel selection (nonoverlapping)



(b) IEEE 802.11b European Channel selection (nonoverlapping)

Obr. 13. Volba nepřekrývajících se frekvenčních pásem (kanálů) v standardu IEEE 802.11b. [10]

IEEE 802.11b

FCC (Federal Communications Commission) v roce 1995 přidělila několik pásem bezdrátového spektra pro bezlicenční použití; jedno z nich je také pro technologické zařízení. V roce 1997 byl ratifikován standard IEEE 802.11, který je již nyní překonán standardem z roku 1999 IEEE 802.11b, který poskytuje maximální teoretickou rychlost 11 Mbit/s v ISM pásmu 2,4 GHz. [10]

IEEE 802.11g

V roce 2003 byl ratifikován standard IEEE 802.11g s maximální teoretickou rychlostí 54 Mbit/s v ISM pásmu 2,4 GHz. Síla signálu klesá v závislosti na vzdálenosti. Kvůli zeslabení signálu při průchodu překážkou a rušení v daném pásmu je při správě připojení automaticky nastavována nižší rychlost (54/48/36/24/12/9/6 Mbit/s). Pokud jsou dva klienti (802.11b a 802.11g) připojeni na 802.11g směrovač, tj. má klient 802.11g nižší přenosovou rychlost. Mnohé směrovače dokáží kombinovat klienty 802.11b/g.

Standard 802.11g je zpětně kompatibilní se standardem 802.11b. V dnešní době je standard 802.11g značně rozšířen. [10]

Rušení

V bezlicenčním pásmu 2,4 GHz je kvůli neexistujícímu omezení možné rušení v podobě elektromagnetické interference (EMI), rušení ve společném kanále (CCI), také známá jako „crosstalk“, rušení v sousedním kanále (ACI), rušení mezi symboly (ISI), rušení mezi nosnými (ICI), způsobené Dopplerovým posunem, nesymetrické rušení (CMI) a rušení vedením.

Z těchto důvodů je potřeba, aby byl průmyslový standard pro bezdrátovou komunikaci robustní a musí spolehlivě překonat možné rušení. K překonání tohoto problému se proto používá metoda přenosu FHSS, jejíž princip spočívá v přeskakování mezi několika frekvencemi při přenosu jednoho bitu nebo několika bitů. Většina průmyslových standardů se k této technice uchyluje z důvodu vyšší spolehlivosti, bezpečnosti a robustnosti. [10]

5 VIZUALIZACE

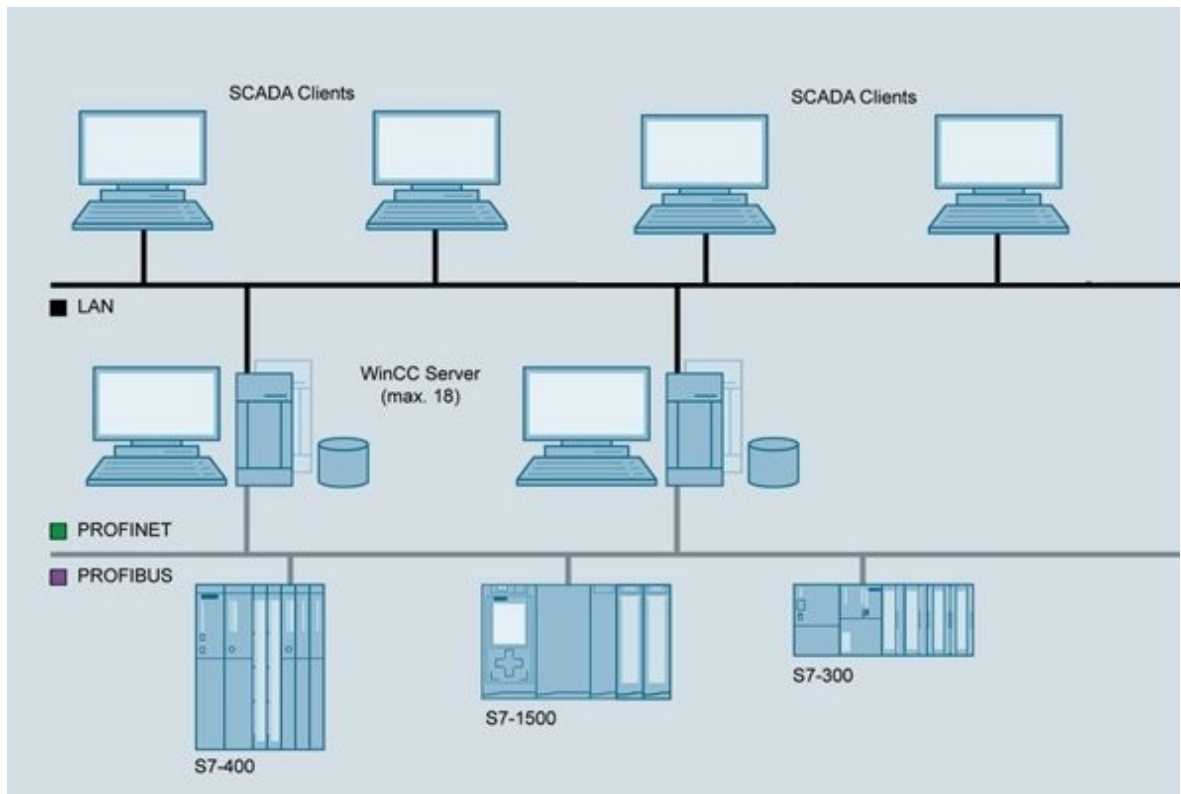
5.1 SCADA

SCADA je zkratka pro Supervisory Control And Data Acquisition tedy „dispečerské řízení a sběr dat“ je široce používán k dohledu a vzdálenému řízení procesů v průmyslovém prostředí.

Výhody SCADA systémů jsou zvláště ceněny u procesů, které jsou velmi rozsáhlé – stovky nebo tisíce kilometrů od jednoho konce k jinému, a to současně se snížením ceny rutinních návštěv při kontrole operačního zařízení.

Běžný systém SCADA je kombinací čtyř různých podsystémů:

1. Hlavní počítač (host system).
2. Místní zařízení pro vzdálené měření a řízení nazývané vzdálený terminál (RTU).
3. Široká oblast komunikačních systémů (telefonní, Internet, intranet) k propojení všech zařízení. Typicky optická komunikační síť je používána k začlenění do páteřních rozvodů.
4. Operátorské rozhraní (HMI), které poskytuje uživatelům přístup k systému. [10]



Obr. 14. Jednoduchá topologie systému SCADA. [35]

Hlavní funkce systému SCADA jsou:

1. Sběr dat z periferních zařízení přes RTU.
2. Zpracování dat za účelem alarmů a jiných významných procesních změn.
3. Poskytování konzistentní databáze procesních informací o zařízení.
4. Prezentace dat přes snadno pochopitelné uživatelské grafické rozhraní s uvedením alarmů, trendů a reportů.
5. Umožnění vzdáleného řízení zařízení ve výrobě.
6. Monitorování systému a jeho diagnostika.
7. Archivování historických dat pro krátkodobé a dlouhodobé uchování a analýzu.
8. Přenos real-time dat do expertních systémů.
9. Poskytování informací pro manažerský informační systém (MIS) a systém pro řízení dodavatelského řetězce (SCM). [10]

5.2 OPC

Velké množství komerčně dostupných SCADA systémů je dnes sestaveno z hardwarových a softwarových komponent od různých výrobců s hotovými interními komponenty, umožňujícími výměnu dat (často přes LAN nebo WAN), a to přes OPC. [10]

OLE for Process Control (OPC) je standard pro výměnu dat, který definuje komunikační rozhraní při řízení a monitorování technologických procesů

OPC se skládá z následujících komponent:

- OPC server – čte data ze zařízení nebo zapisuje data do zařízení.
- OPC klient – přistupuje k datům z OPC serveru. [15]

6 DATABÁZOVÉ SYSTÉMY

SŘBD (systém řízení báze dat) nebo database-management system (DBMS) je soubor nezávislých dat a sada programů sloužících k přístupu k těmto datům.

Mezi hlavní výhody patří účelný a efektivní způsob uložení a vyhledání informace. Databázové systémy jsou navrženy tak, aby zvládly pojmout velké množství informací. Správa dat zahrnuje definování struktur pro uložení informace a poskytuje prostředky pro manipulaci s informacemi.

Databázový systém musí navíc zajistit bezpečnost uložených informací navzdory systémovým výpadkům nebo pokusům o neautorizovaný přístup. Pokud jsou data sdílena mezi několika uživateli, musí systém zabránit možnému nestandardnímu chování. Jelikož jsou informace důležité v mnoha společnostech, došlo ke vzniku velkého množství konceptů a technik pro správu dat. [14]

Jazykové prostředky SŘBD:

- Jazyk pro definici dat (DDL) – slouží k vytvoření všech definic uživatelských dat potřebných v aplikaci
- Jazyk pro manipulaci dat (DML) – používá se k aktualizaci dat (přidávání, odstraňování a změny dat v databázi)
- Příkazy pro provoz a údržbu databáze (DCL) – spravuje uživatelské role a práva [9]

V dnešní době se uplatňují především relační databáze. Jako základní jazyk relačních systémů se používá SQL, který zahrnuje nejen DDL a DML ale i další podjazyky. SQL je zkratka anglického termínu pro strukturovaný dotazovací jazyk. Jedná se o nejběžnější způsob komunikace s databázovými servery a tento jazyk podporují téměř všechny databázové systémy. [11]

6.1 PostgreSQL

PostgreSQL je relační databázový systém s otevřeným zdrojovým kódem a je založený na objektově-relačním modelu. Je vydán pod licencí PostgreSQL License (podobná BSD či MIT). Systém vyvíjí komunita vývojářů a firem. Běží na všech rozšířených operačních

systemech: GNU/Linux, unixových (AIX, BSD, HP-UX, SGI-IRIX, Mac OS X, Solaris, Tru64) a Windows. Stoprocentně splňuje podmínky ACID, plně podporuje cizí klíče, operace JOIN, pohledy, triggerů a uložené procedury. Obsahuje většinu SQL-92 a SQL-99 datových typů, např. INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL a TIMESTAMP. Nechybí ani podpora moderních datových typů jako je JSON nebo XML. [23]

6.1.1 Rozhraní pro správu databáze – pgAdmin

pgAdmin je free a open-source grafické administrační rozhraní pro PostgreSQL podporované většinou běžných platform. První prototyp, nazvaný pgManager, byl napsán pro PostgreSQL 6.3.2 v roce 1998, a po několika měsících přepsán a vydán pod GPL licenci jako pgAdmin. Aktuální verze je pgAdmin 4, původně vydaná pod Artistic License a nyní je vydávána pod stejnou licenci jako PostgreSQL. [24]

6.2 SQLite

SQLite je relativně malá knihovna, která implementuje kompletní relační databázový systém. Nejedná se však o klasickou klient-server databázi tak, jak u PostgreSQL, ale databáze se ukládá do jednoho souboru a předpokládá se tedy, že k datům přistupuje software, který běží na stejném počítači či serveru, kde jsou uložena data. [26]

Základní myšlenkou SQLite je jednoduchost. SQLite nevyžaduje složitou konfiguraci. Samozřejmě je nutné, aby uživatel, pod kterým poběží software přistupující k datům, měl k databázovému souboru náležitá práva. Neznamená to však, že SQLite nelze vůbec konfigurovat. Chování instance SQLite lze upravit pomocí příkazu PRAGMA. SQLite lze použít na všech distribucích GNU/Linux, Windows i mac OS X. Z uvedeného vyplývá, že možnosti použití SQLite jsou velmi rozmanité a multiplatformní. [25]

Výhodami SQLite jsou jednoduchost použití, malá velikost, databáze v souboru, většinou nulová potřeba jakékoli konfigurace, nulová cena a volné užití. Jako další výhodou je také nutno uvést i jistou míru paralelního přístupu, kterou, na rozdíl od konkurenčních projektů, SQLite nabízí. SQLite umožňuje, aby z jedné databáze četlo najednou více instancí. Toto

však neplatí pro zápis. Další výhodou může být uložení databáze v jednom jediném souboru a tak je její zálohování a případně i distribuce snadná.

Nevýhodou je nízký výkon, který se projeví při zápisu velkého množství dat. SQLite chápe každý INSERT jako vlastní transakci a podle dokumentace transakci dokončí až když jsou data uložena na disku. Nevýhodou je také již uzavření celé databáze v případě probíhajícího zápisu. Nelze tedy realizovat paralelní zápis do databáze a také číst z databáze, pokud současně probíhá zápis. [26]

7 BEZPEČNOST V PRŮMYSLOVÉ AUTOMATIZACI

Jedním z problémů, kterým musí čelit profesionálové v oblasti automatizace je fakt, že řídicí systémy v továrnách (ICS) a SCADA systémy jsou navrženy k otevřenosti, robustnosti, snadnému zacházení a opravám, ale nejsou už tolik zabezpečeny. Z několika důvodů již nejsou automatizované systémy od informačních systémů izolovány a to přináší další hrozby. [10]

Současný trend zvyšování konektivity, vzájemné spolupráce a účinnost technologií, které jsou používány v podnicích, se týkají také ICS. Původně izolované systémy jsou nyní více závislé na propojení s externími technologiemi. To vede k tvorbě nových zranitelností, které jsou znatelnou hrozbou pro ICS. Z tohoto důvodu je nezbytné věnovat velké úsilí pro analýzu zranitelností. Zanedbání této oblasti může vést ke zničení nebo nedostupnosti ICS. [20]

II. PRAKTICKÁ ČÁST

8 VOLBA PROGRAMOVACÍHO JAZYKA

Cílem práce bylo získávání dat z PLC SIMATIC firmy Siemens, s tím, že data jsou ukládána do databázového systému, odkud jsou přebírána k vizualizaci, analýze a grafické prezentaci sledovaných parametrů. Zařízení musí být levné, nasaditelné v průmyslovém prostředí.

Výsledným řešením je systém pro sběr a analýzu dat, který zvládne obsluhovat i nekvalifikovaný personál.

8.1 Volba programovacího jazyka

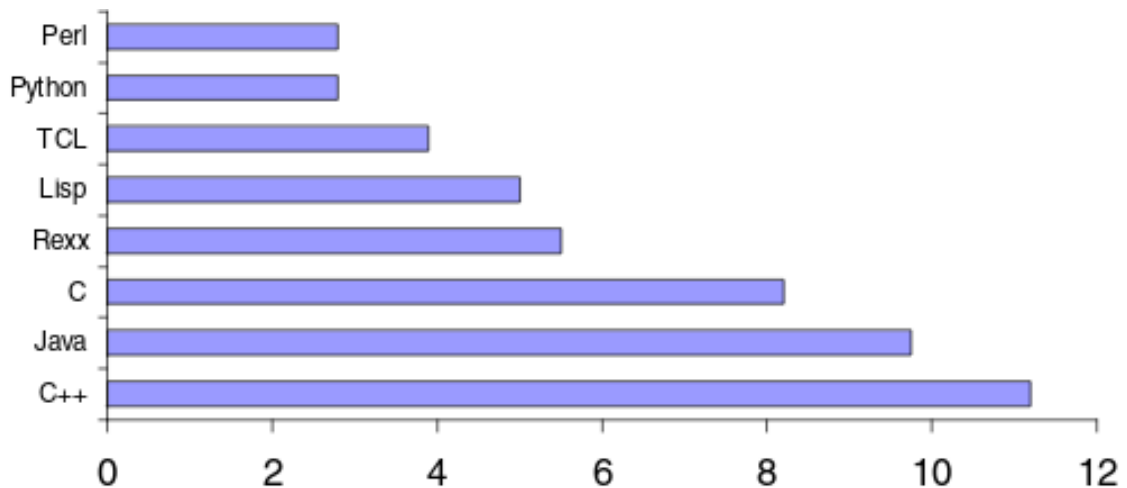
V tomto projektu bylo třeba zvolit vhodný programovací jazyk. Uživatelé, kteří budou řešení využívat, mohou mít malé zkušenosti s programováním (elektrotechnici, robotici). Z tohoto důvodu je potřeba zvolit jazyk jednoduchý na naučení, srozumitelný tak, aby si obsluha mohla v budoucnu kód sama přizpůsobovat.

Kritéria pro výběr programovacího jazyka jsou zásadní a vhodnost programovacího jazyka lze posuzovat z několika pohledů. [19]

Dle TIOBE indexu se mezi nejpoužívanější programovací jazyky v poslední době řadí: C, C++, Java, JavaScript, PHP, Python, Ruby, SQL, C#, Ruby, Visual Basic .NET. [27]

Co se týká snadnosti na naučení, umísťuje se jazyk Python na předních příčkách, syntaxe jazyka je čitelná a zároveň má silnou komunitu, velké množství dostupných návodů.

Pro tento projekt je také důležitá multiplatformnost a nasaditelnost na OS Windows a Linux, což Python plně splňuje. Nevýhodou Pythonu zůstává rychlost při zpracování programu, může být ale výborný při prototypování, kdy je doba od zadání k realizaci projektu velmi krátká. [28]



Obr. 15. Průměrný čas (v hodinách) na vyřešení problému. [29]

8.1.1 Python

Autorem multiparadigmatického vysokoúrovňového programovacího jazyka Python je Guido van Rossum. Základní cíle jazyka jsou:

- Snadnost a intuitivnost
- Otevřenost kódu
- Srozumitelnost kódu
- Vhodnost pro každodenní úkoly, umožňující vývoj v krátkém čase

Python je vhodný k datové analýze, má mnoho knihoven pro vizualizaci dat a používá se také v oblasti testování, pro vývoj webových aplikací a také vědecké výpočty. Je to univerzální skriptovací jazyk běžící na každém PC. Slouží k rychlému prototypování. [21]

Pro vizualizaci dat se jako nejvhodnější jeví jazyk Python. Má mnoho knihoven pro vizualizaci, jako jsou Matplotlib, Saeborn, Mpld3 podporuje 2D a 3D grafy.

Kromě standardních knihoven jsou k dispozici tisíce knihoven třetích stran které jsou snadno dostupné na repozitáři PyPi, přes správce modulů pip nebo pip3, který je součástí běžné instalace Pythonu. Tento správce umožňuje jejich – odinstalaci, upgrade, správu verzí.

9 KOMUNIKACE S PLC SIMATIC

Pro komunikaci s PLC přes Ethernet nebo Wi-Fi lze využít jeden z mnoha na trhu dostupných OPC serverů. Jejich nevýhodou je, že jsou licenčně vázané a zkušební verze mají značná omezení. [30]

System pro sběr dat doporučovaný společností Siemens je SCADA systém SIMATIC WinCC v kombinaci s OPC serverem SIMATIC NET SOFTNET-IE S7. Oficiální návod popisuje spojení s PLC S7-1200. Při tomto řešení je ale vyžadováno napsání programu v PLC ve STEP 7, což je pro potřebu tohoto řešení nevhodné.

Další možností je napsání samostatného komunikačního rozhraní, neboť PLC S7 disponuje funkcemi pro komunikaci. [31]

Existuje také řešení, které není licenčně vázané a je založeno na open-source projektu LibNoDave [32], který je obsažen v knihovně Snap7, která zastupuje OPC server a umožňuje komunikovat s PLC pomocí proprietárního protokolu S7.

Bezplatné systémy nesplňují současně podmínky jednoduchosti, snadné modifikovatelnosti

9.1 Konfigurace PLC Simatic

PLC SIMATIC mají modulární architekturu. Jádrem každé sestavy je jednotka procesoru (CPU), ke které lze připojovat další moduly pomocí integrované sběrnice. Každá řada PLC (S7-300, S7-400, S7-1200 a S7-1500) používá svoji vlastní a nelze tedy jednotlivé komponenty z různých řad zaměňovat. Z pohledu hardwarové konfigurace je sběrnice označována jako rack. Fyzická realizace racku je buď ve formě montážní lišty se samostatnými propojovacími konektory nebo jako montážní šasi s integrovanými konektory. V rámci jednoho projektu je PLC sestaveno v jednom racku (pořadové číslo 0), ale může jich být i více. Množství racků závisí na požadavcích zákazníka, na fyzických dispozicích rozváděče a na úrovni decentralizace řešení.

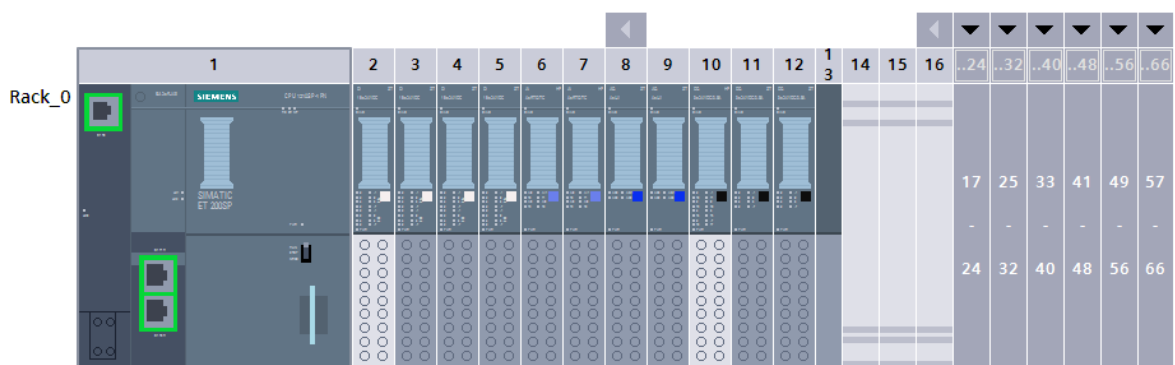
Hardwarovou konfiguraci definuje projektant nebo programátor při instalaci zařízení. Pro řady S7-300 a S7-400 k tomu slouží softwarový nástroj SIMATIC Manager – HW Config.

Slot	Module	Order number	Firmware	MPI address	I address	Q address	Comment
1							
2	CPU 318-2	6ES7 318-2AJ00-0AB0	V1.2	2			
X2	DP-Master				8191*		
X1	MPI/DP			2	8190*		
3							
4	CP 343-1 TCP	6GK7 343-1EX00-0XE0	V5.0		256...271	256...271	
5							
6							
7							
8							
9							
10							
11							

Obr. 16. Příklad hardwarové konfigurace PLC řady S7-300 (Simatic Manager).

Slot	Module	Order number	Firmware	MPI address	I address	Q address	Comment
1	PS 407 10A	6ES7 407-0KR00-0AA0			8191*		
3	CPU 414-3 DP	6ES7 414-3XJ00-0AB0	V1.2				
X2	DP-master				8190*		
X1	MPI/DP				8189*		
IF1	PROFIBUS_IF	6ES7 964-2AA01-0AB0			8140*		
5	CP 443-1	6GK7 443-1EX11-0XE0	V1.1		8188*		
6							
7							
8							
9							

Obr. 17. Příklad hardwarové konfigurace PLC řady S7-400 (Simatic Manager).



Obr. 18. Vizualizace hardwarové konfigurace CPU 1510SP-1 PN (TIA Portal).

Z obrázků (Obr. 16, 17 a 18) je možné vyčíst označení čísla racku (ve všech příkladech 0), umístění CPU, komunikačního procesoru (CP) a v případě S7-400 také umístění

napájecího modulu (PS). Každá pozice (slot) ovlivňuje rozsah vnitřních vstupních (I address) a výstupních adres (Q address) příslušného modulu. Pro navázání komunikace s CPU prostřednictvím komunikačního procesoru (CP) je důležité znát IP adresu CP a masku sítě. Tu lze zjistit anebo nastavit v dialogovém okně CP (Object properties). Pro navázání komunikace z vnější sítě (Industrial Ethernet, PROFINET) je také důležité znát umístění CPU v rámci racku. V předchozích případech jsou to pozice 2 (CPU 318-2), 3 (CPU 414-3 DP) a 1 (CPU 1510SP-1 PN).

9.2 STEP 7

SIMATIC používá jako standardní programovací jazyk STEP 7, který slouží k tvorbě programů pro PLC v žebříkovém diagramu LAD, jazyku funkčních bloků (FBD) nebo seznamu instrukcí (STL). LAD je vhodný například pro elektrikáře a připomíná elektrický obvod, STL pro uživatele ze světa počítačové technologie – připomíná jazyk symbolických adres. 67[4]

9.3 Snap7

Snap7 je open-source, 32 i 64bitový multiplatformní komunikační modul umožňující propojení s PLC Siemens SIMATIC S7. Je nezávislý na platformě a v současné době podporuje operační systémy Windows, GNU/Linux, BSD, Oracle Solaris 11, mac OS X. Vhodný je také pro malé na Linuxu založené ARM nebo MIPS desky jako je Raspberry Pi.

Snap7 podporuje širokou škálu programovacích jazyků, jako jsou C /C++, .NET/Mono, Pascal, LabVIEW, Python nebo Node.js.

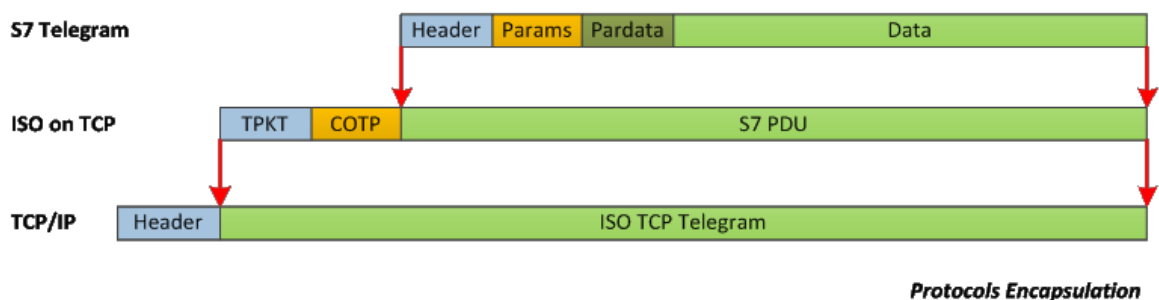
Snap7 je navržen pouze pro komunikaci přes Ethernet. Ten má oproti Profibus/MPI několik výhod. Je to například rychlost (1000 Mb/s), nižší cena a jednoduchost při řešení problémů. Komunikační procesor (CP) PLC zajišťuje komunikaci přes Ethernet a využívá k tomu dva protokoly TCP/IP a S7 Protokol. [40]

S7 Protokol

S7 Protokol, nebo také ISO TCP (RFC1006) je základním protokolem pro komunikaci zařízení společnosti Siemens, který je založený na blokové struktuře. Jednotlivé bloky se nazývají PDU (Protocol Data Unit) a jejich maximální délka závisí na CP a je upravována v průběhu připojení. Každý přenos obsahuje příkaz nebo odpověď na něj. Pokud se data nevejdou do jednoho PDU, pak musejí být rozděleny mezi několika dalších PDU.

Každý příkaz obsahuje hlavičku, sadu parametrů, data parametrů a datový blok. Hlavička a sada parametrů jsou vždy přítomny, zbylé dva elementy jsou volitelné.

S7 Protocol, ISO TCP a TCP/IP respektují zapouzdření: každý telegram je datovou částí nižšího protokolu. [40]



Obr. 19. Zapouzdření protokolů. [40]

9.3.1 Navázání spojení pomocí Snap7

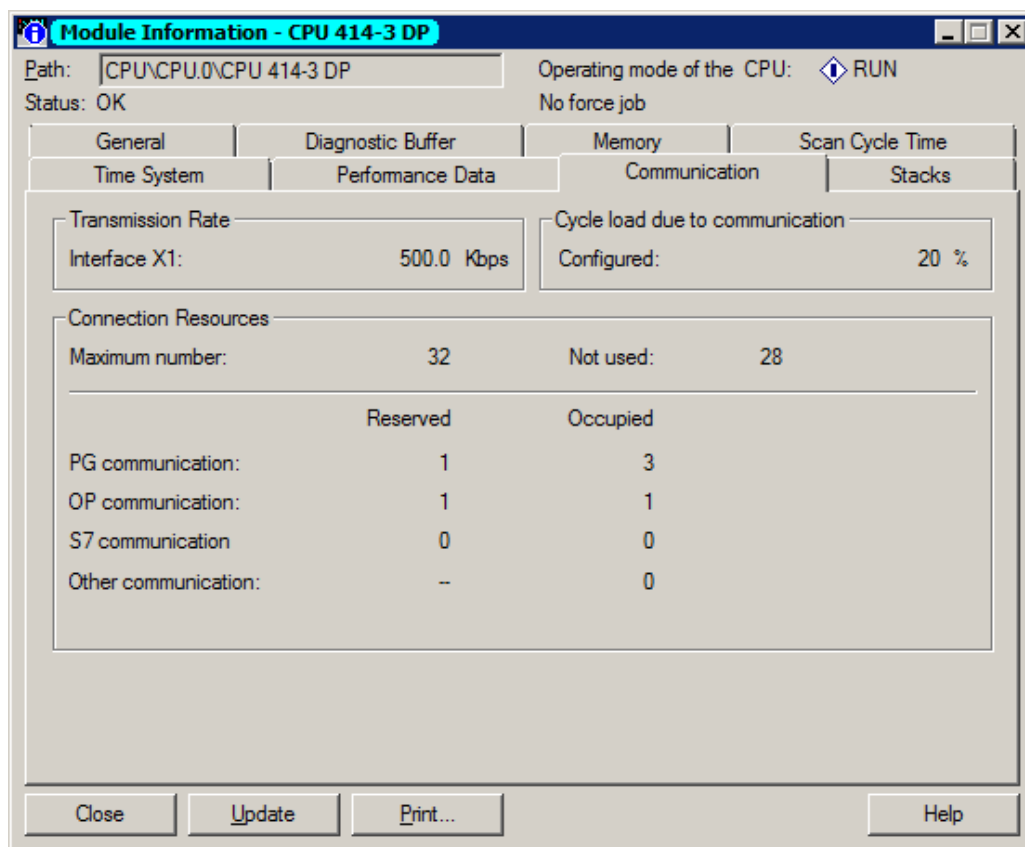
Před navázáním spojení s PLC je nutné brát v potaz jistá omezení. Jsou jimi například maximální počet dostupných zdrojů (spojení) nebo délka cyklu běhu programu. Pochopením těchto zákonitostí se lze vyhnout některým problémům. Například nemá smysl se dotazovat na data vícekrát během jednoho cyklu.

9.3.1.1 Počet dostupných spojení

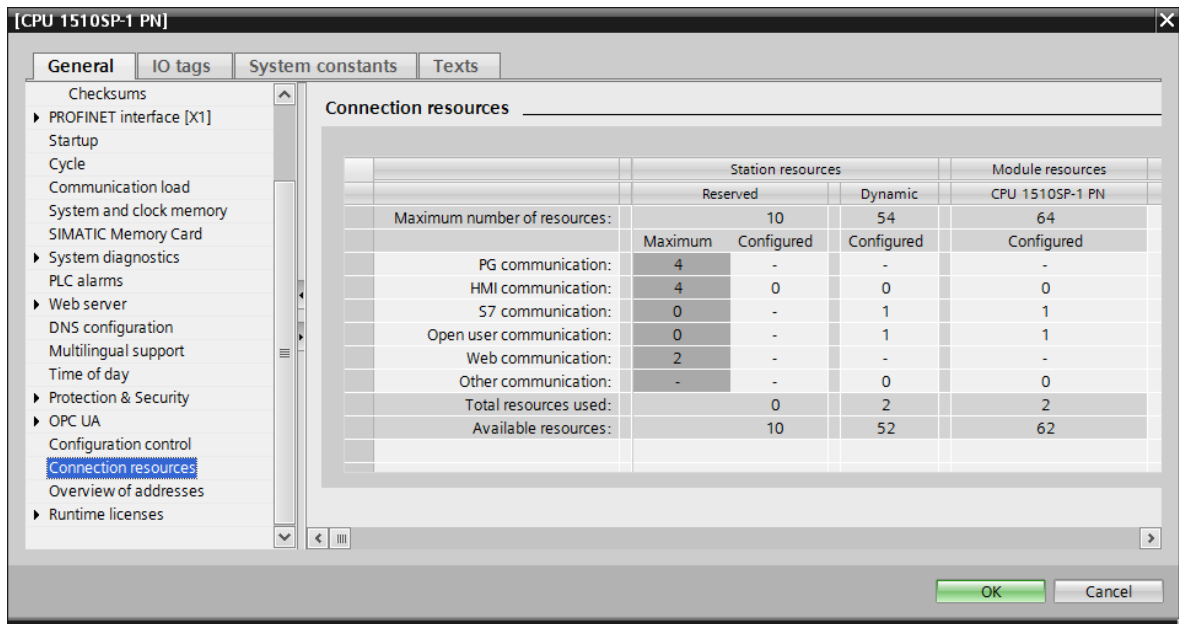
Každé síťové spojení vyžaduje rezervaci datových struktur v paměti, proto je jejich maximální počet omezený. Každý model CPU disponuje jiným počtem dostupných spojení. Tato hodnota je dostupná v dokumentaci. Navíc ne každé spojení lze použít pro libovolný typ zařízení. Spojení jsou rezervována pro PG komunikaci, HMI nebo

komunikaci s webovým serverem; nelze je zaměňovat. Dostupná spojení jsou závislá na konkrétním CPU a CP (komunikačním procesoru). [37]

Karta Communication (Obr. 20) dialogového okna Module Information (Simatic Manager) informuje o přenosových rychlostech rozhraní, poměru zátěže CPU dostupné pro komunikaci, počtu možných a obsazených spojení prostřednictvím rozhraní Industrial Ethernet. Tabulka Connection Resources je dostupná pouze když je CPU v módu RUN. Komunikace prostřednictvím Snap7 se u S7-400 vyhodnocuje jako PG communication (na Obr. 20 jsou vidět 3 aktivní spojení).



Obr. 20. Dialogové okno Module Information – Communication (S7-400).

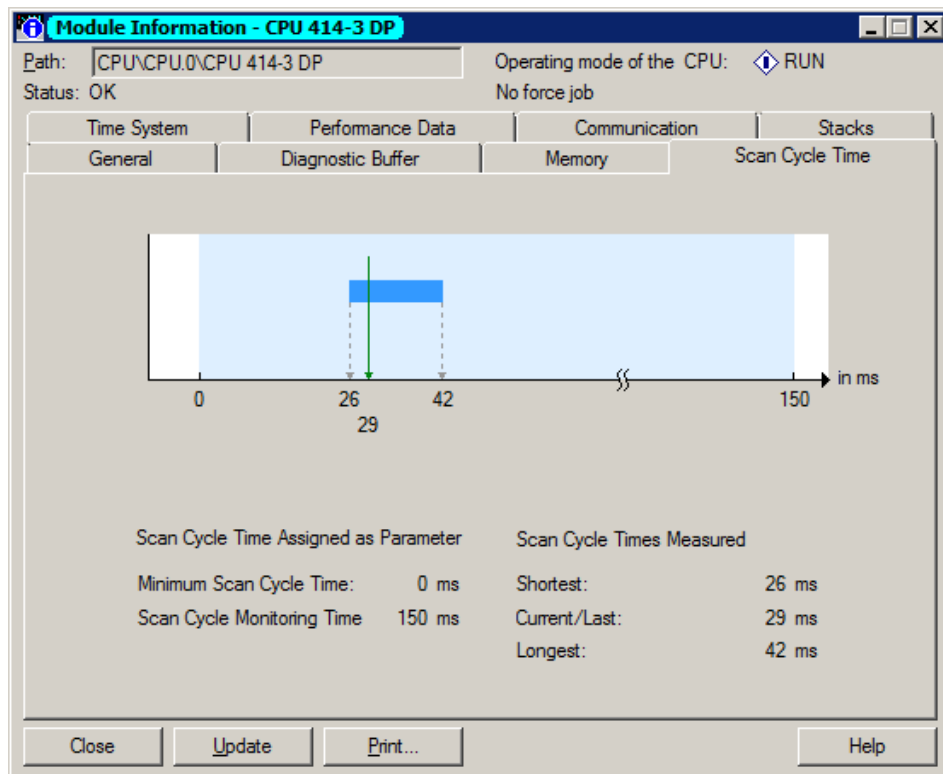


Obr. 21. Dialogové okno Module Information – Communication (S7-1500).

9.3.1.2 Délka cyklu

Běh hlavního programu je neustále monitorován, délka jeho cyklu je popsána veličinou cycle monitoring time. Pokud zpracování programu trvá déle než je stanovená mez, CPU zavolá organizační blok OB80 Time Error. Pokud není blok přítomen, chyba je při prvním výskytu ignorována. Výchozí limit je stanovený na 150 ms, ale tuto hodnotu lze změnit. [37]

Karta Scan Cycle Time (Obr. 22) dialogového okna Module Information (Simatic Manager) poskytuje informace o délce cyklu, tj. době jednoho průchodu uživatelského programu u zvoleného CPU. Naměřené časy jsou graficky znázorněny barevnými značkami na ose. Délka osy je dána stanovenou maximální délkou cyklu. Na ose se zobrazují tyto hodnoty: Délka nejkratšího, nejdelšího a posledního cyklu od poslední změny z módu STOP do módu RUN. Časy, které jsou mimo stanovené meze se zobrazují červeně.



Obr. 22. Dialogové okno Module Information – Scan Cycle Time.

9.3.1.3 Snap7 klient

Z pohledu S7 komunikace patří mezi klienty například tyto knihovny a nástroje: LibNoDave, ProDave, SAPI-S7, SIMATIC NET a navzdory svému názvu také kterýkoliv OPC Server. Třída Snap7Client knihovny Snap7 se chová také jako klient. Podporuje plně protokol S7; lze číst, zapisovat, provádět operace (download, upload), ovládat PLC (RUN, STOP, provádět kompresi...) a nastavovat hesla.

Instalace Snap7

Snap7 je možné instalovat na platformách GNU/Linux stažením zdrojového kódu a jeho zkompilováním. Pro distribuci Ubuntu je k dispozici instalační repozitář, ze kterého lze zdrojový kód stáhnout a potom provést instalaci běžným způsobem pomocí programu apt-get.

```
# instalace snap7 z repozitáře
$ sudo add-apt-repository ppa:gijzelaar/snap7
$ sudo apt-get update
$ sudo apt-get install libsnap71 libsnap7-dev
```

```
# instalace python-snap7 z PyPI (Python Package Index)
$ pip install python-snap7
```

Pro operační systémy Windows je nutné stáhnout knihovnu snap7.dll, která je k dispozici na stránce <http://sourceforge.net/projects/snap7/files/> a uložit ji buď do adresáře C:\Windows\System32\ nebo kamkoliv jinam, ale pak je nutné uvést tuto cestu do proměnné prostředí PATH.

Instalace Python wrapperu python-snap7 je společná pro všechny operační systémy, a to pomocí příkazu pip, který je součástí instalace Python 3.

Navázání spojení se realizuje importem modulu (wrapperu) snap7, vytvořením instance třídy snap7.client.Client() a zavoláním její metody connect(). Parametry metody jsou: IP adresa CP (komunikačního procesoru), číslo racku a číslo slotu.

```
import snap7
import time

spojeni = snap7.client.Client()
spojeni.connect('192.168.0.100', 0, 3)

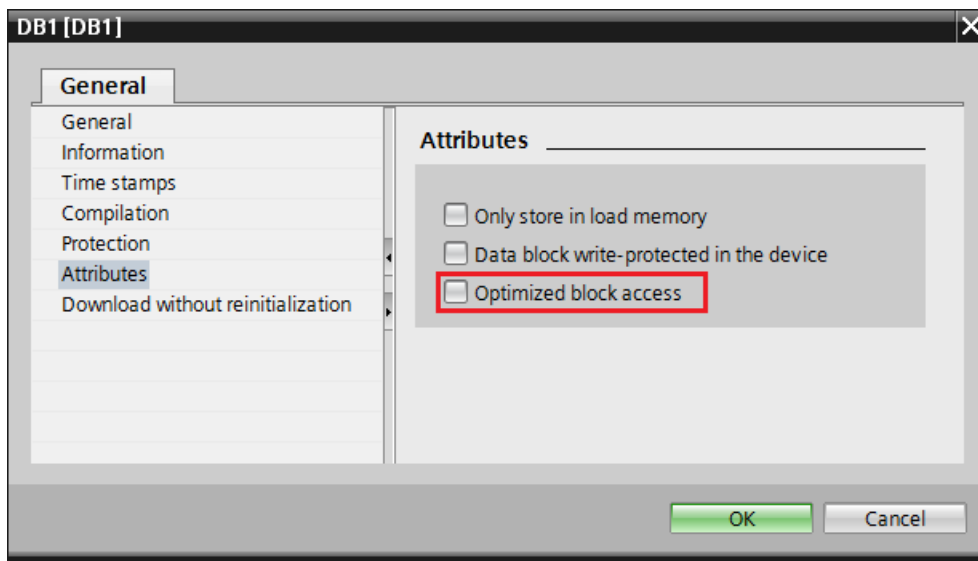
while True:
    # vlastní komunikace
    time.sleep(1) # časový krok smyčky

spojeni.disconnect()
```

Specifika CPU S7-1200 a S7-1500

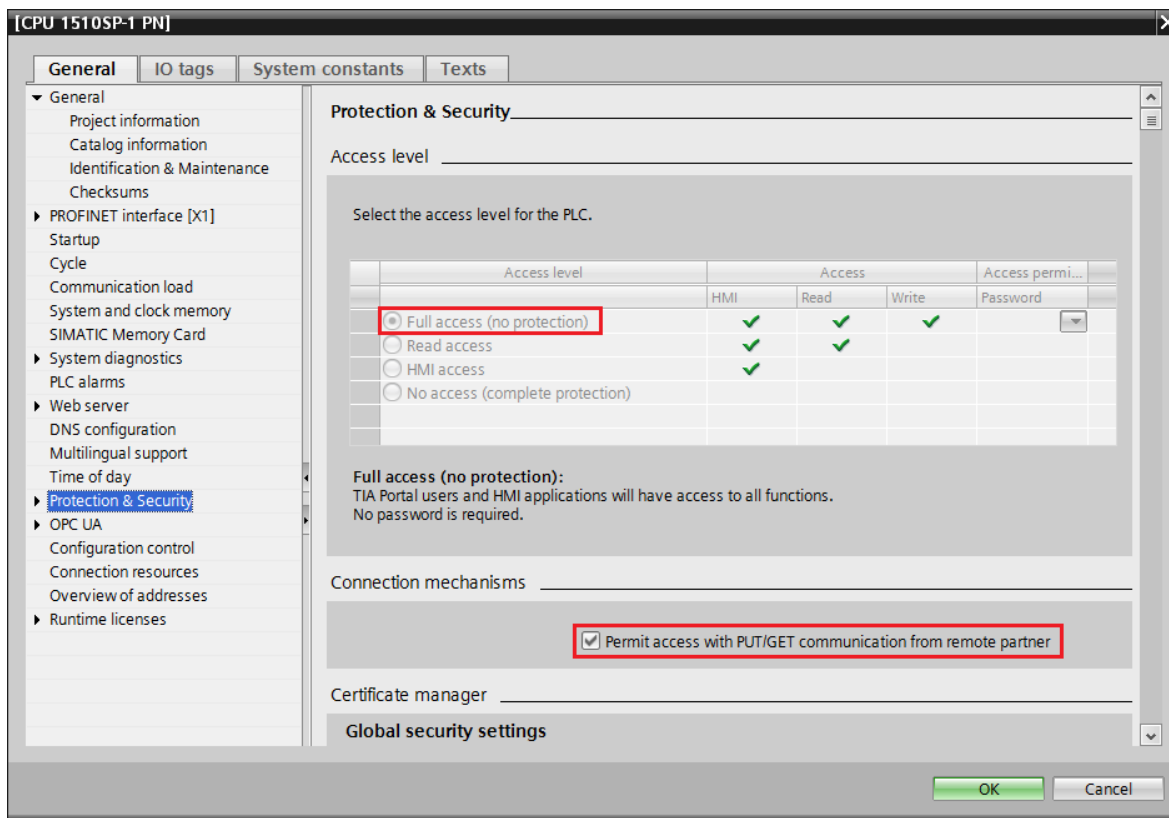
Pomocí Snap7 lze komunikovat s CPU S7-1200 a S7-1500 pomocí S7 protokolu, ale spojení se identifikuje jako HMI zařízení, takže je povolen přenos pouze základních datových struktur. Řídící funkce nebo práce s datovými bloky nejsou v současnosti knihovnou Snap7 podporovány. Dostupné jsou pouze globální datové bloky (DB) a to za těchto podmínek:

Musí být vypnuta funkce „Optimized block access“ (Obr. 22). Dialogové okno lze zobrazit pomocí kontextového menu a volbou „Properties“.



Obr. 23. Vypnutí funkce „Optimized block access“.

Úroveň zabezpečení musí být nastavena na „Full access“ (Obr. 23). Menu je dostupné v nastavení CPU, volba „Properties“. Kromě plného přístupu je nutné povolit „Permit access with PUT/GET communication from remote partner (PLC, HMI, OPC, ...)“.



Obr. 24. Nastavení úrovně ochrany.

9.3.2 Čtení a interpretace dat

Datové typy určují formu reprezentace obsahu datových oblastí. STEP 7 poskytuje několik předdefinovaných typů, které jsou globálně dostupné (v celém projektu) a mohou být použity ve kterémkoliv bloku. [37]

Dále se budu zabývat jen těmi datovými typy, které jsou pro řešení podstatné. Mezi základní datové typy patří BOOL, BYTE, WORD, DWORD a REAL.

BOOL reprezentuje hodnotu o velikosti jeden bit (tj. hodnotu „0“ nebo „1“, FALSE nebo TRUE), datový typ BYTE obsazuje 8 bitů, WORD 16 bitů a DWORD 32 bitů. Základním typem pro reprezentaci reálného čísla je REAL. [37]

Stavy signálů nebo číselné hodnoty mají v PLC přímo přiřazené adresy v oblasti operandů. Například operand I2.5 obsahuje ID operandu (I), adresu (2) a spolu s bitovým operátorem také adresu bitu (5). Bity jsou počítány zprava doleva od nuly. [37]

Adresy vstupů (I) a výstupů (Q) z periférií jsou definovány hardwarovou konfigurací.

Datový operand je operand uvnitř datového bloku. Například DB10.DBW4 znamená adresu 4 datového typu WORD v datovém bloku 10. Maximální počet datových bloků je závislý na použitém CPU. [37]

Tab. 3. Přehled nejběžnějších operandů.

Operand area	Operand ID	Bit (1 bit)	Byte (8 bits)	Word (16 bits)	Doubleword (32 bits)
Input	I	%Iy.x	%IBy	%IWy	%IDy
Peripheral input	The input operand is expanded with :P	%Iy.x:P	%IBy:P	%IWy:P	%IDy:P
Output	Q	%Qy.x	%QBy	%QWy	%QDy
Peripheral output	The output operand is expanded with :P	%Qy.x:P	%QBy:P	%QWy:P	%QDy:P
Bit memory	M	%My.x	%MBy	%MWy	%MDy

y = byte address; x = bit address

Tab. 4. Identifikátory jednotlivých oblastí ve Snap7.

identifikátor	hodnota	význam
snap7.types.S7AreaPE	0x81	procesní vstupy
snap7.types.S7AreaPA	0x82	procesní výstupy
snap7.types.S7AreaMK	0x83	merkery
snap7.types.S7AreaDB	0x84	DB
snap7.types.S7AreaCT	0x1C	čítače
snap7.types.S7AreaTM	0x1D	časovače

Na základě těchto poznatků lze snadno zapsat volání metod pro čtení z PLC. Níže jsou na jednotlivých řádcích uvedeny příklady čtení z různých datových oblastí a v komentáři je uveden příslušný PLC operand.

```
spojeni.read_area(snap7.types.S7AreaPE, 0, 686, 2) # IW686
spojeni.read_area(snap7.types.S7AreaDB, 199, 14, 2) # DB199.DBW14
spojeni.read_area(snap7.types.S7AreaMK, 0, 1000, 2) # MW1000
```

Endianita

Funkce `read_area` vrací hodnotu typu `bytearray` a ve formátu `big-endian`. PC mají zpravidla formát `little-endian`. Je proto nutné bajty interpretovat ve správném pořadí. K tomu byly použity dvě různé funkce. V případě celočíselných hodnot bylo možné použít metodu třídy `int` – `from_bytes`, která umožňuje endianitu zdroje specifikovat:

```
int.from_bytes(vstup, byteorder='big', signed=False)
```

V případě reálných čísel byly použity funkce `pack/unpack` z modulu `struct`:

```
struct.unpack('>f', struct.pack('4B', *vstup))[0]
```

kde symbol „>“ znamená formát `big-endian`, „f“ typ `float` a „4B“ délku 4 bajty. Operátor * (hvězdička) provede dekompozici proměnné `vstup` na tuple.

Binární hodnoty

Pro zápis jednotlivých bitů byl zvolen typ `str` pro snazší ukládání do souborů CSV. Funkce vrací řetězec symbolů '0' a '1' typu `str`, které je možné jednotlivě indexovat pomocí hranatých závorek. Index se shoduje s indexem v PLC. Pro potřebu načítání až 32bitových

typů je funkce dimenzována i pro tyto případy. Index pak ale nekoresponduje s indexem v PLC:

```
def bity(vstup):  
    '''Převod bytearray byte na retězec bitů 0 a 1'''  
    cislo = bin(int.from_bytes(vstup, byteorder='big', signed=False))  
    return (cislo[2:].zfill(32)) [::-1]
```

9.4 Hostitelské PC

Aplikace může běžet na jakémkoliv počítači, kterým může být IPC, PAC, kancelářské PC nebo také Raspberry Pi. Řešení bylo testováno na běžném PC s OS MS Windows, GNU/Linux a na Raspberry Pi 3, model B s OS Raspbian. Výhodou nasazení na Raspberry Pi je jeho cena, snadná dostupnost, možnost jednorázového nasazení, ale není však stavěno pro vyšší odolnost v průmyslovém prostředí (extrémní teploty, vlhkost). Dnešní průmyslové sítě jsou ale tak rozvětvené i do mimoprámyslových prostorů, takže není nutné instalovat zařízení přímo v provozu. Lze také použít jakékoliv PC se síťovou kartou, kde je nainstalovaný MS Windows nebo GNU/Linux.

9.4.1 Raspberry Pi 3 model B

Pro testování aplikace byl použit mikropočítač Raspberry Pi 3 B: Broadcom BCM2837 CPU 4× ARM Cortex-A53, 1,2 GHz, RAM: 1 GB LPDDR2 (900 MHz), 10/100 Ethernet, 2,4 GHz 802.11n wireless, Bluetooth 4.1 Classic, microSD 8 GB; 4× USB 2.0, Ethernet.

Na mikropočítač byla nainstalovaná aplikace a ve spolupráci s pracovníkem údržby byla přidělena IP adresa. Mikropočítač byl připojen ethernetovým kabelem do odbočky páteřní sítě prostřednictvím průmyslového switchu SCALANCE XC-208. Ve spolupráci s programátorem PLC byl definován seznam adres, ze kterých byla sledovaná data čtena.



Obr. 25. PLC SIMATIC s Raspberry Pi 3, model B.

10 NÁVRH DATABÁZOVÉHO SYSTÉMU

Kritériem pro výběr databázového systému bylo open-source řešení, dále také snadná instalace, konfigurace a samozřejmě multiplatformnost (aplikovatelné na počítačích s operačními systémy GNU/ Linux a Windows). Dle srovnání používaných databázových systémů (Obr. 25) byla zvolena řešení s volnou licencí: PostgreSQL a SQLite. Pro řešení bylo zvažováno také použití systému MariaDB.

342 systems in ranking, May 2018

Rank	Rank			DBMS	Database Model	Score		
	May 2018	Apr 2018	May 2017			May 2018	Apr 2018	May 2017
1.	1.	1.		Oracle +	Relational DBMS	1290.42	+0.63	-63.90
2.	2.	2.		MySQL +	Relational DBMS	1223.34	-3.06	-116.69
3.	3.	3.		Microsoft SQL Server +	Relational DBMS	1085.84	-9.67	-127.96
4.	4.	4.		PostgreSQL +	Relational DBMS	400.90	+5.43	+34.99
5.	5.	5.		MongoDB +	Document store	342.11	+0.70	+10.53
6.	6.	6.		DB2 +	Relational DBMS	185.61	-3.34	-3.23
7.	↑9.	↑9.		Redis +	Key-value store	135.35	+5.24	+17.90
8.	↓7.	↓7.		Microsoft Access	Relational DBMS	133.11	+0.89	+3.24
9.	↓8.	↑11.		Elasticsearch +	Search engine	130.44	-0.92	+21.62
10.	10.	↓8.		Cassandra +	Wide column store	117.83	-1.26	-5.28
11.	11.	↓10.		SQLite +	Relational DBMS	115.45	-0.53	-0.61
12.	12.	12.		Teradata	Relational DBMS	74.41	+0.74	-1.91
13.	13.	↑16.		Splunk	Search engine	65.09	+0.04	+8.40
14.	14.	↑18.		MariaDB +	Relational DBMS	64.99	+0.44	+14.01
15.	15.	↓14.		Solr	Search engine	61.51	-1.70	-2.26
16.	16.	↓13.		SAP Adaptive Server +	Relational DBMS	61.51	-0.12	-6.24
17.	17.	↓15.		HBase +	Wide column store	59.95	+0.26	+0.44
18.	18.	↑20.		Hive +	Relational DBMS	56.97	-0.43	+13.49
19.	19.	↓17.		FileMaker	Relational DBMS	54.67	-0.33	-1.81
20.	20.	↓19.		SAP HANA +	Relational DBMS	48.37	-0.52	-0.68

Obr. 26. Srovnání používaných databázových systémů. [33]

Tab. 5. Přehled použitých databázových systémů.

Databáze	Licence	Informace
PostgreSQL	The PostgreSQL Licence	relační databáze, zdarma i pro komerční účely, možnost serverového řešení
SQLite	Public Domain Open-Source	relační databáze, zdarma i pro komerční účely, každá databáze v samostatném souboru

Databázový systém SQLite je nativně podporován mnoha programovacími jazyky. Je integrován i v Python 3 (modul `sqlite.py`) a použití systému SQLite zejména na Raspberry Pi se nabízí vzhledem k nízkým nárokům na velikost databázového systému (knihovna má přibližně 700 KiB) a také kvůli snadnému zálohování, protože je databáze uložena v jednom souboru.

10.1 SQLite

10.1.1 Instalace

Pro příklad je uvedena instalace na OS Raspbian, což je operační systém založený na Debian GNU/Linux. Instalace je tedy stejná jako ve všech distribucích založených na distribuci Debianu či Ubuntu:

```
$ sudo apt install sqlite3
```

SQLite knihovna je vybavena jádrem Shell SQLite. Databáze se vytvoří příkazem:

```
$ sqlite3 db.sqlite3
```

kde `db.sqlite3` je název souboru, ve kterém je databáze uložena. Po spuštění příkazu se otevře příkazový interpret, pomocí kterého lze ručně zadávat příkazy SQL.

Instalace modulu `sqlite3` pro Python není nutná, protože je v každé výchozí instalaci přítomný.

10.1.2 Datové typy

V každém záznamu tabulky je nutné uvést přesný čas odečtené hodnoty. Nabízejí se dvě možnosti, jak čas uložit. Buď relativně (od daného okamžiku) nebo absolutně ve formě lokálního nebo světového času. První varianta může být efektivnější pro úsporu místa v databázi, ale znehledňuje dodatečnou interpretaci dat. A právě kvůli lepší čitelnosti dat byl zvolen lokální čas.

SQLite nemá speciální třídu pro ukládání data a času. Namísto toho jsou interní funkce schopné pracovat s typy TEXT, REAL a INTEGER. Podporovaný zápis pro TEXT je ISO8601 formát "YYYY-MM-DD HH:MM:SS.SSS". Zápis do typu REAL počítá dny od

roku 4714 př. n. l. Celočíslný INTEGER je zápis unixového času, tedy počet sekund od 1970-01-01 00:00:00 UTC. Pro řešení byl zvolen typ TEXT. Jedná se, jak již bylo uvedeno, o textový řetězec, který není úsporný, ale přehledný. TEXT podporuje kódování UTF-8 (též UTF-16BE a UTF-16LE), což není pro zápis data relevantní.

Pro celočíselné hodnoty je k dispozici datový typ INTEGER. Jedná se o celočíselný typ se znaménkem. Velikost záznamu je 1, 2, 3, 4, 6 nebo 8 bajtů a závisí na velikosti ukládané hodnoty. Do tohoto datového typu je také možné ukládat logické hodnoty jako 0 a 1, přestože SQLite disponuje datovým typem NUMERIC, který je pro proměnné typu BOOLEAN preferovaný. Reálná čísla jsou ukládána v datovém typu REAL, který respektuje standard IEEE 754 pro dvojkovou aritmetiku v pohyblivé řádové čárce.

Při pořízení více dat v jednom časovém okamžiku se nabízí možnost ukládat všechna tato data do jednoho záznamu v tabulce. V případě pořízení celočíselné (t1), reálné (t2) a logické (t3) hodnoty lze vytvořit databázovou tabulku (např. sber20180521) SQL příkazem CREATE TABLE. Abychom zabránili chybě při vytváření tabulky s duplicitním názvem, je vhodné případnou existující tabulku odstranit příkazem DROP.

```
DROP TABLE IF EXISTS sber20180521;
CREATE TABLE sber20180521(cas TEXT, t1 INTEGER, t2 REAL, t3 INTEGER);
```

Zápis dat do databáze může probíhat buď přímo během pořizování dat nebo je možné data dodatečně importovat z CSV:

```
import sqlite3
import csv

conn = sqlite3.connect('db.sqlite3')      # připojí se k databázi
cur = conn.cursor()                      # vytvoří kurzor

# cur.execute('DROP TABLE IF EXISTS sber20180521;') # smazání tabulky
sql = 'CREATE TABLE sber20180521(cas TEXT, t1 INTEGER, t2 REAL, t3
INTEGER);'
cur.execute(sql)                          # provedení SQL příkazu

with open('sber20180521.csv') as csvfile:
    reader = csv.reader(csvfile, delimiter=';')
    sql = 'INSERT INTO sber20180521(cas, T1, T2, T3) VALUES (?, ?, ?, ?);'
    next(reader)                          # vynechá hlavičku CSV souboru
    for row in reader:                    # prochází každý řádek
        cur.execute(sql, row)             # provede INSERT

conn.commit()
conn.close()
```

10.2 PostgreSQL

Instalace PostgreSQL na operačních systémech GNU/Linux založených na distribuci Debian se spustí příkazem:

```
$ sudo apt-get install postgresql
```

Pokud se budou využívat i další nástroje, jako je klient nebo webová aplikace pgAdmin, je nutné doinstalovat i specifické knihovny (libpq-dev):

```
$ sudo apt-get install libpq-dev postgresql-client postgresql-client-  
common  
$ sudo apt install pgadmin3 # u novějších distribucí (2018-04) pgadmin4
```

Aby bylo možné vytvářet databázové uživatele, je nutné se přepnout na uživatele postgres:

```
$ sudo su postgres
```

Pokud se vytvoří uživatel se stejným jménem jako je ten systémový, budou mu automaticky přidělena práva k databázi. V případě výchozího uživatele v systému Raspbian se nabízí uživatel pi.

```
$ createuser pi -P --interactive
```

Následují dotazy na vytvoření hesla, zda má být uživatel superuser, a pokud ne, pak jaká mají být uživateli přidělena práva.

Jako adaptér pro připojení k databázi PostgreSQL byl zvolen Psycopg, který je k dispozici na PyPI (Python Package Index) i v předkompilované verzi (psycopg2-binary):

```
$ pip3 install psycopg2-binary
```

Protože moduly sqlite3 i psycopg2 splňují specifikaci Python DB API 2.0, která vyžaduje podobnost (částečnou kompatibilitu) modulů zajišťujících spojení s databázemi, lze uvést jen zásadní rozdíly v zápisu kódu:

```
import psycopg2 # import modulu psycopg2  
  
# možnosti vytvoření nového spojení (connection object)  
conn = psycopg2.connect(dbname='sber', user='pi', password='hEsLo')  
conn = psycopg2.connect('dbname=sber user=pi password=hEsLo')
```


Základními parametry spojení jsou: dbname (název databáze), user (uživatel), password (heslo), host (adresa serveru) a port (číslo portu, výchozí je 5432).

Psycopg převádí proměnné Pythonu na typy SQL hodnot a naopak. Není tedy problém použít příkaz:

```
cur.execute("""INSERT INTO tabulka (xdate, xint, xstr) VALUES (%s, %s, %s);""", (datetime.date(2018, 05, 21), 13, "SIMATIC"))
```

Chování konkrétní adaptace je nutné vyčíst z dokumentace. [38]

10.2.1 Datové typy

Na rozdíl od minimalistického SQLite disponuje PostgreSQL velmi širokou škálou datových typů. Pro celočíselné byly zvolen typ INTEGER. Logické hodnoty lze zapisovat různými způsoby, stejně jako u SQLite. Výchozím PostgreSQL datovým typem pro logické hodnoty je BOOLEAN. Pro reálná čísla byl zvolený typ REAL. Jak je vidět, názvy zvolených datových typů se neliší od těch, které byly zvoleny u SQLite. To umožňuje kompatibilita systémů na úrovni standardu SQL-92.

Pro zápis data je v PostgreSQL možné využít datový typ TIME. I zde ale lze zapsat hodnotu jako TEXT, který zajistí kompatibilitu s CSV i SQLite, ale zkomplikuje provádění případných výpočtů na úrovni databáze.

11 VIZUALIZACE ZÍSKANÝCH DAT

Ve vizualizační části bylo pro výstup zvoleno webové rozhraní a to z důvodu zobrazení na PC a mobilním zařízení. Jazyk Python podporuje mnoho frameworků pro tvorbu webového prostředí (např. Bottle, CherryPy, Django, Flask, Pyramid, Tornado). Pro tento projekt byl zvolen mikroframework Flask. Kód je možný implementovat a použít i na jiném frameworku, který podporuje Python 3.

11.1 Flask

Webový mikroframework Flask je distribuován pod BSD licenci a podporuje Python 2 a Python 3. Obsahuje vestavěný server WSGI. Flask je navržen tak, aby práce probíhala rychle a jednoduše s možností převedení na komplexní aplikace.

- vestavěný server a debugger pro vývoj
- integrovaná jednotka pro podporu testování
- RESTful (webová služba) obsluhování požadavků
- šablonovací systém Jinja2
- podpora secure cookies (na straně klienta)
- Werkzeug WSGI toolkit
- Podpora unicode
- rozsáhlá dokumentace [36]

Instalace Flask na Windows:

```
pip install flask
```

Instalace Flask na GNU/Linux:

```
$ pip3 install flask
```

11.2 Knihovny Pythonu

Pandas – knihovna pro zpracování dat která umožňuje jejich snadnější manipulaci

Matplotlib – knihovna pro zpracování grafů

Mpld3 (Matplotlib, D3js) – knihovna pro vykreslování grafů

Mpld3 spojuje hlavní knihovnu Pythonu – Matplotlib pro tvorbu grafů a JavaScript knihovnu D3 k tomu, aby zobrazení v prohlížeči bylo co nejvíce příjemné k použití. Výhodou použití matplotlib s Mpld3 je přívětivé grafické rozhraní umožňující interakci uživatele s grafem, nevýhodou je, že knihovna využívá JavaScript, což zpomaluje vykreslování grafu při velkých objemech dat.

Výše uvedené knihovny jsou k dispozici na PyPI (Python Package Index) a instalace na operačních systémech GNU/Linux je následující:

```
$ pip3 install <název příslušného balíčku>
```

Instalace balíčků na operačním systému Windows je podobná:

```
pip install <název příslušného balíčku>
```

11.3 Vizualizovaná data

Přes webové rozhraní je možné vizualizovat data načtená z databázových systémů SQLite, PostgreSQL a také jako další zdroj dat byl zprovozněn modul pro načtení dat z CSV, který umožní vizualizovat i dříve nasbíraná data. K nahrání souboru CSV z klienta slouží formulář, který zasílá metodou POST proměnné z webového formuláře na webový server, kde se zpracovávají příslušným skriptem.

The screenshot shows a web application interface with a dark blue background. At the top, there is a navigation bar with the text 'PLC Data Acquisition Analysis System' on the left and 'INDEX', 'SBĚR DAT', 'VIZUALIZACE', and 'NÁPOVĚDA' on the right. Below the navigation bar, the main heading is 'Nahrát csv soubor'. Underneath this heading, there are two buttons: 'Vybrat soubor' and 'Upload'. Below the buttons, there is a section titled 'Zadej zdroj dat' with three radio button options: 'SQLite', 'PostgreSQL', and 'CSV'. At the bottom of this section, there is a 'Zobrazit' button.

Obr. 27. Formulář pro nahrání souboru CSV a volba zdroje dat.

```
@app.route('/vizualizace.html', methods=['GET', 'POST'])
def vizualizace():
    """Formulář pro výběr zdroje a přesměrování ke zpracování"""

    if request.method == 'POST':
        volba_zdroje = request.form['zdroj']

        if volba_zdroje == 'db':                # SQLite
            return redirect(url_for('db'))
        elif volba_zdroje == 'dbpg':           # PostgreSQL
            return redirect(url_for('dbpg'))
        elif volba_zdroje == 'csv':            # CSV
            return redirect(url_for('csv'))
    return render_template('vizualizace.html')
```

Dále se pomocí dekorátoru zajistí, že návratová hodnota funkce dbpg() bude obsahem cesty /dbpg.html (zpracování dat z PostgreSQL databáze)

```
@app.route('/dbpg.html')
```

Zpracování dat probíhá tak, že se skript připojí ke zdroji dat (SQLite, PostgreSQL nebo CSV). SQL dotaz zažádá o celou datovou sadu z databázové tabulky. V případě CSV

souboru čtení zajišťuje metoda `read_csv` z modulu `pandas`. Data se následně převedou do struktury `DataFrame`.

```
def dbpg():
    """zpracování dat - PostgreSQL"""

    try:
        conn = psycopg2.connect(dbname='', user='', host='localhost',
password='')
        print('connecting...', conn)
    except:
        error = "Chyba spojení s databází."
        print(error)
        return render_template('vizualizace.html', error=error)
    cur = conn.cursor()
    sql = 'SELECT cas, T1, T2, T3, T4 as hodnoty from zaznamy ORDER BY
cas;'
    cur.execute(sql)

    pole_hodnot = []
    for row in cur:
        pole_hodnot.append([row[0], float(row[1]), float(row[2]),
float(row[3]), float(row[4])])

    df = pd.DataFrame(pole_hodnot, columns=['cas', 'T1', 'T2', 'T3',
'T4'])
    cur.close()
```

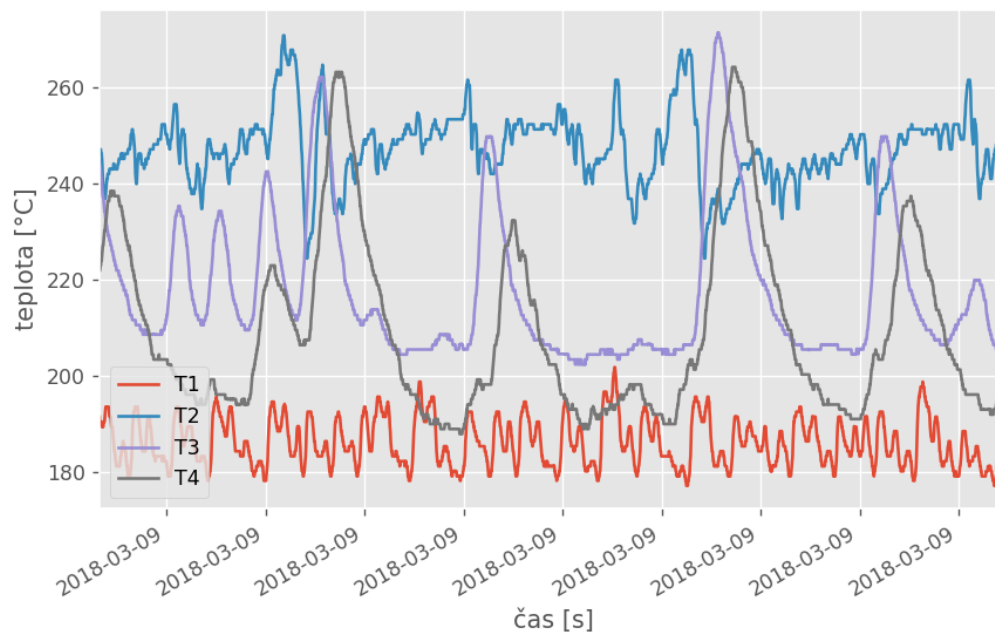
Vykreslení probíhá prostřednictvím knihovny `matplotlib` a modulu `Mpld3`, pomocí které se graf stává interaktivním. Je možné nastavit parametry jako je velikost grafu, popisky os anebo například formátování data.

```
# definice stylu a obsah grafu
plt.style.use('ggplot')
fig, ax = plt.subplots(figsize=(8, 5))
df.plot(ax=ax, x='cas', y=['T1', 'T2', 'T3', 'T4'])
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
plt.gcf().autofmt_xdate()
plt.ylabel('teplota [°C]')
plt.xlabel('čas [s]')
plt.legend()

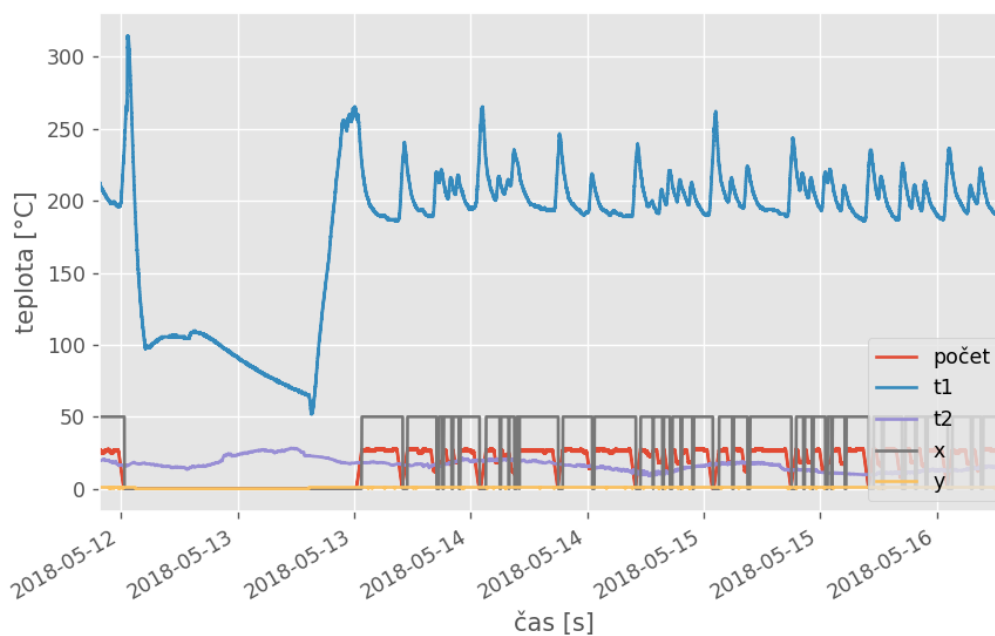
data_dbpg = mpld3.fig_to_html(fig, no_extras=True)
```

Jako výstup je dále vypsána statistika s uvedením počtu záznamů nejnižší, nejvyšší a průměrnou teplotou, dalšími údaji jsou pak začátek a konec měření. Statistiku je dále možné do budoucna rozšířit o další požadované hodnoty

```
# statistika
pocet_mereni = len(df.index)
```

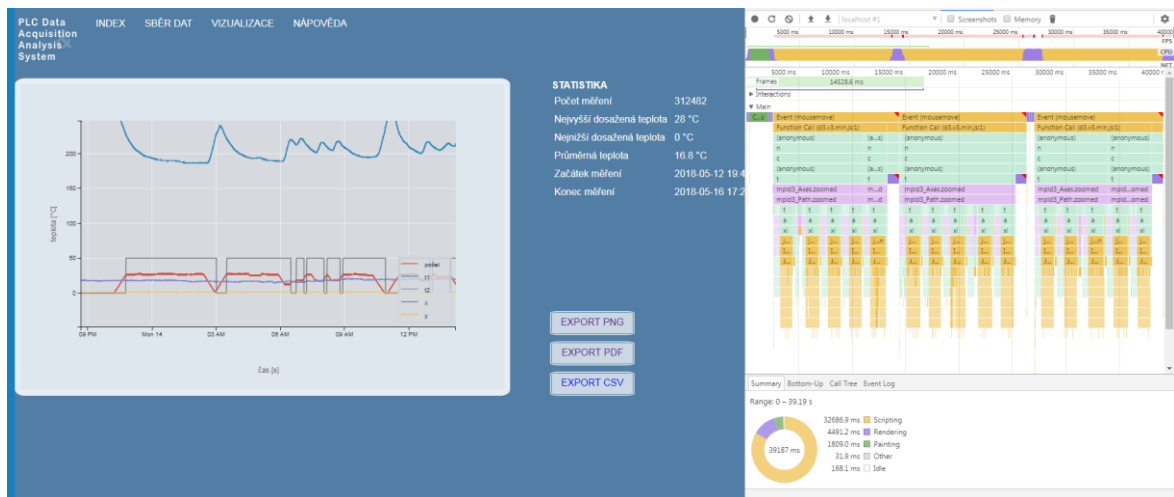



Obr. 29. Ukázka exportu do formátu PNG.



Obr. 30. Ukázka exportu do formátu PNG.

Bylo vyzkoušeno i načtení většího množství dat, přibližně 300 000 řádků. Toto velké množství vstupů se již vykresluje poměrně dlouho. Pomalé vykreslení je způsobeno především JavaScriptem, jak je vidět ze statistiky internetového prohlížeče.



Obr. 31. Vykreslený graf s výkonnostní charakteristikou prohlížeče.

Profilování

Pro testování výkonnostních charakteristik byl použit modul cProfile, který poskytuje podrobný rozpis počtů a časů volání. Modul ukazuje přesně, co se volá a jak dlouho každé volání trvá. [21]

```
import profilePerformance
import cProfile, pstats, io
```

```
if __name__ == '__main__':
    pr = cProfile.Profile()
    pr.enable()
    profilePerformance.db()
    pr.disable()

    s = io.StringIO()
    ps = pstats.Stats(pr, stream=s)
    ps.strip_dirs().sort_stats('cumulative').print_stats()
    print(s.getvalue())
```



```

connecting... <sqlite3.Connection object at 0x00000000064C5D50>
1032227 function calls (1027138 primitive calls) in 6.538 seconds

Ordered by: cumulative time

ncalls  tottime  percall  cumtime  percall  filename:lineno(function)
   1    0.382    0.382    6.538    6.538  profilePerformance.py:128(db)
   1    0.000    0.000    1.882    1.882  plotting.py:3758(__call__)
   1    0.000    0.000    1.882    1.882  plotting.py:2624(plot_frame)
   1    0.000    0.000    1.881    1.881  plotting.py:2416(_plot)
   1    0.004    0.004    1.861    1.861  plotting.py:1039(generate)
   1    0.015    0.015    1.635    1.635  plotting.py:1167(_post_plot_logic_common)
   1    0.086    0.086    1.545    1.545  plotting.py:1169(<listcomp>)
   1    0.000    0.000    1.243    1.243  frame.py:1303(to_csv)
   1    0.000    0.000    1.242    1.242  format.py:1452(save)
59743  0.266    0.000    1.242    0.000  printing.py:156(pprint_thing)
   1    0.001    0.001    1.227    1.227  format.py:1560(_save)
   3    0.005    0.002    1.226    0.409  format.py:1578(_save_chunk)
   3    0.000    0.000    1.059    0.353  figure.py:1470(savefig)
   3    0.000    0.000    1.059    0.353  backend_bases.py:2087(print_figure)
   1    0.000    0.000    0.948    0.948  pyplot.py:1056(subplots)
   1    0.001    0.001    0.889    0.889  _display.py:172(fig_to_html)
   1    0.000    0.000    0.824    0.824  pyplot.py:428(figure)
   1    0.000    0.000    0.824    0.824  backend_tkagg.py:75(new_figure_manager)
   1    0.000    0.000    0.823    0.823  backend_tkagg.py:84(new_figure_manager_given_figure)
321/3  0.002    0.000    0.740    0.247  artist.py:61(draw_wrapper)
   3    0.000    0.000    0.740    0.247  figure.py:1090(draw)
   6/3  0.000    0.000    0.735    0.245  image.py:120(_draw_list_compositing_images)

```

Obr. 32. Příklad výpisu funkce cProfile.

Na základě měření výkonnosti je možné vysledovat slabiny systému a cíleně je optimalizovat. Z měření (Obr. 32) je zřejmé, že největší slabinou testovaného kódu je vykreslování grafu (plotting.py). Obecně se ukázalo, že generování grafických výstupů pomocí externích knihoven způsobuje největší potíže z pohledu výkonnosti.

ZÁVĚR

V této práci bylo navrženo řešení pro sběr a analýzu dat z PLC SIMATIC firmy Siemens. Bylo popsáno, jak se data z PLC SIMATIC v současnosti získávají a také bylo navrženo řešení s postupem, který pro tyto účely využívá open-source knihovnu Snap7. Byly zvoleny vhodné databázové systémy, a to SQLite a PostgreSQL. Řešení také současně počítá s výměnou dat prostřednictvím CSV souborů.

Vizualizace sledovaných parametrů je realizována za pomoci webového frameworku Flask. Aplikace je programována v jazyku Python. Výstupem pro nasbíraná data je interaktivní graf a základní statistika. Součástí výstupu jsou také exporty do souborů PNG, PDF a CSV. Navržené řešení poskytuje programátorům PLC možnost sledování systému v době jeho ožívování, a také poskytnout pracovníkům údržby rychlou diagnostiku poruch a možnost snížit doby prostojů během úprav nebo odstraňování problémů.

Byla podrobně popsána implementace jednotlivých kroků včetně instalace jednotlivých knihoven, programů, modulů a nástrojů.

Jedním z cílů bylo zpřístupnění sběru dat i méně kvalifikovanému personálu. Ti ale budou vždy, zejména při prvotním nastavení a oživení, odkázáni na asistenci pracovníka znalého problematiky adresování v PLC.

Do budoucna je možné celou aplikaci vylepšit o prvky jako je vizualizace dat v reálném čase, podrobnější statistickou analýzu s možností následné predikce poruch pro potřebu údržby, nebo lze vytvořit rozhraní pro uživatelskou konfiguraci (GUI). V řešení vůbec nebyly uvažovány PLC jiných výrobců. Také lze v budoucnu implementovat podporu jiných databázových systémů.

SEZNAM POUŽITÉ LITERATURY

- [1] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace, 1.díl. 1. Vyd. Praha: BEN – technická literatura, 1999. ISBN 80-860-5658-9.
- [2] ŠMEJKAL, Ladislav. PLC a automatizace, 2.díl. 1. Vyd. Praha: BEN – technická literatura, 2005. ISBN 80-7300-087-3.
- [3] MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty. Vyd. 2. Praha: Vydavatelství ČVUT, 2004. ISBN 80-010-2925-5.
- [4] SIMATIC Working with STEP 7 V5.5 (Getting Started) [online]. Norimberk: Siemens AG, 2010 [cit. 2017-01-30]. Dostupné z:
<https://support.industry.siemens.com/cs/document/45531551/>
- [5] SIMATIC Programming with STEP 7 V5.5 (Programming Manual) [online]. Norimberk: Siemens AG, 2010 [cit. 2017-01-30]. Dostupné z:
<https://support.industry.siemens.com/cs/document/45531107/>
- [6] SIMATIC NET Industrial Ethernet/PROFINET Industrial Ethernet (System Manual) [online]. Norimberk: Siemens AG, 2016 [cit. 2017-01-30]. Dostupné z:
<https://support.industry.siemens.com/cs/document/27069465/>
- [7] BERGER, Von Hans. Automatisieren mit STEP 7 in AWL: speicherprogrammierbare Steuerungen SIMATIC S7-300/400. 2., wesentlich überarb. und erw. Aufl. Erlangen: Publicis-MCD-Verl, 1998. ISBN 38-957-8089-8.
- [8] KABELOVÁ, Alena a Libor DOSTÁLEK. Velký průvodce protokoly TCP/IPa systémem DNS. 5., aktualiz. vyd. Brno: Computer Press, 2008, 488 s. ISBN 978-80-251-2236-5.
- [9] POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. Praha: České vysoké učení technické v Praze, 2013. ISBN 978-80-01-05212-9.
- [10] REDDY, Y Jaganmohan. *Industrial process automation systems*. Boston, MA: Elsevier, 2014. ISBN 978-0-12-800939-0.
- [11] MOMJIAN, Bruce. *PostgreSQL: praktický průvodce*. Brno: Computer Press, 2003. ISBN 80-7226-954-2.
- [12] *Trends in advanced intelligent control, optimization and automation*. New York, NY: Springer Berlin Heidelberg, 2017. ISBN 978-3-319-60698-9.

- [13] PUŽMANOVÁ, Rita. *TCP/IP v kostce*. České Budějovice: Kopp, 2004. ISBN 80-7232-236-2.
- [14] SILBERSCHATZ, Abraham., Henry F. KORTH a S. SUDARSHAN. *Database system concepts*. 6th ed. New York: McGraw-Hill, c2011. ISBN 978-0-07-352332-3.
- [15] ZEŽULKA, František. *Prostředky průmyslové automatizace*. Brno: VUTIUM, 2004. ISBN 80-214-2610-1.
- [16] *Dynamics in logistics*. New York, NY: Springer Berlin Heidelberg, 2016. ISBN 978-3-319-45116-9.
- [17] *Data mining and big data*. New York, NY: Springer Berlin Heidelberg, 2016. ISBN 978-3-319-40972-6.
- [18] WINN, Michael, Mason RICE, Stephen DUNLAP, Juan LOPEZ a Barry MULLINS. Constructing cost-effective and targetable industrial control system honeypots for production networks. *International Journal of Critical Infrastructure Protection* [online]. 2015, **10**, 47-58 [cit. 2018-05-09]. DOI: 10.1016/j.ijcip.2015.04.002. ISSN 18745482. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S1874548215000256>
- [19] PARKER, Kevin R., Thomas A. OTTAWAY a Joseph T. CHAO. Criteria for the selection of a programming language for introductory courses. *International Journal of Knowledge and Learning* [online]. 2006, **2**(1/2), 119- [cit. 2018-05-13]. DOI: 10.1504/IJKL.2006.009683. ISSN 1741-1009. Dostupné z: <http://www.inderscience.com/link.php?id=9683>
- [20] VÁVRA, Jan, Martin HROMADA a Roman JAŠEK. Specification of the Current State Vulnerabilities Related to Industrial Control Systems. *International Journal of Online Engineering (iJOE)* [online]. 2015, **11**(5), 64- [cit. 2018-05-14]. DOI: 10.3991/ijoe.v11i5.4981. ISSN 1861-2121. Dostupné z: <http://online-journals.org/index.php/i-joe/article/view/4981>
- [21] SUMMERFIELD, Mark. *Python 3: výukový kurz*. Brno: Computer Press, 2010. ISBN 978-80-251-2737-7.
- [22] Vrstvy průmyslové sítě. In: <Http://www07.abb.com> [online]. [cit. 2018-05-20]. Dostupné z: http://www07.abb.com/images/librariesprovider104/default-album/800xa-network-topology-690_690.jpg

- [23] *PostgreSQL* [online]. 2018 [cit. 2018-05-20]. Dostupné z:
<https://postgres.cz/wiki/PostgreSQL>
- [24] PostgreSQL. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-20]. Dostupné z:
<https://cs.wikipedia.org/wiki/PostgreSQL>
- [25] SQLite. *SQLite* [online]. <http://sqlite.org/>, 2018 [cit. 2018-05-20]. Dostupné z:
<http://sqlite.org/>
- [26] Seznámení s *SQLite*. *Seznámení s SQLite* [online]. <https://blog.root.cz/maertienuv-obcasny-blog/seznameni-s-sqlite/>, 2016 [cit. 2018-05-20]. Dostupné z:
<https://blog.root.cz/maertienuv-obcasny-blog/seznameni-s-sqlite/>
- [27] *TIOBE Index for May 2018* [online]. 2018 [cit. 2018-05-20]. Dostupné z:
<https://www.tiobe.com/tiobe-index/>
- [28] PRECHELT, L. An empirical comparison of seven programming languages. *Computer* [online]. **33**(10), 23-29 [cit. 2018-05-20]. DOI: 10.1109/2.876288. ISSN 00189162. Dostupné z:
<http://ieeexplore.ieee.org/document/876288/>
- [29] Median Hours to Solve Problem. In: *Connelly Barnes* [online]. [cit. 2018-05-20]. Dostupné z: https://www.connellybarnes.com/documents/language_productivity.pdf
- [30] OPC SERVERY PRO SIMATIC S7 - PŘEHLED TRHU [online]. [cit. 2018-05-20]. Dostupné z: <https://www.blaja.cz/archiv-clanku/opc-servery-pro-simatic-s7-prehled-trhu.html>
- [31] Process Data Acquisition and Monitoring with the SIMATIC S7-1200 (Data Logging) [online]. [cit. 2018-05-20]. Dostupné z:
[https://support.industry.siemens.com/cs/document/64396156/process-data-acquisition-and-monitoring-with-the-simatic-s7-1200-\(data-logging\)?dti=0&lc=en-WW](https://support.industry.siemens.com/cs/document/64396156/process-data-acquisition-and-monitoring-with-the-simatic-s7-1200-(data-logging)?dti=0&lc=en-WW)
- [32] Libnodave [online]. [cit. 2018-05-20]. Dostupné z:
<https://sourceforge.net/projects/libnodave/>
- [33] DB-Engines Ranking [online]. In: . [cit. 2018-05-20]. Dostupné z: <https://db-engines.com/en/ranking>
- [34] Nanobox PC [online]. [cit. 2018-05-21]. Dostupné z:
<https://mall.industry.siemens.com/mall/en/us/Catalog/Product/6ES76478BB222BA1>

- [35] WinCC/Server [online]. [cit. 2018-05-21]. Dostupné z:
<https://w3.siemens.com/mcms/human-machine-interface/en/visualization-software/scada/wincc-options/wincc-server/pages/default.aspx>
- [36] *Flask* [online]. [cit. 2018-05-21]. Dostupné z: <http://flask.pocoo.org/>
- [37] BERGER, Hans. *Automating with SIMATIC S7-1500 Configuring, Programming, Motion Control and Security inside TIA Portal*. Erlangen: PUBLICIS, 2013. ISBN 9783895784040.
- [38] *Psycopg – PostgreSQL database adapter for Python* [online]. [cit. 2018-05-21]. Dostupné z: <http://initd.org/psycopg/docs/>
- [39] *SIMATIC S7-1500, ET 200MP Automation system System Manual* [online]. [cit. 2018-05-21]. Dostupné z:
https://cache.industry.siemens.com/dl/files/792/59191792/att_895925/v1/s71500_et200mp_system_manual_en-US_en-US.pdf
- [40] *Snap7* [online]. [cit. 2018-05-21]. Dostupné z: <http://snap7.sourceforge.net/>
- [41] *SIMATIC S7-300* [online]. [cit. 2018-05-21]. Dostupné z:
<https://w3.siemens.com/mcms/industrial-communication/en/profibus/system-interfacing-s7/s7-300/Pages/s7-300.aspx>
- [42] *SIMATIC S7-400* [online]. [cit. 2018-05-23]. Dostupné z:
<https://w3.siemens.com/mcms/programmable-logic-controller/en/advanced-controller/s7-400/Pages/Default.aspx>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ACI	Adjacent-channel interference
ACID	Atomicity, Consistency, Isolation, Durability
CCI	Co-channel interference
CMI	Common-mode interference
EMI	Electromagnetic interference
FBD	Function Block Diagram
FHSS	Frequency Hopping Spread Spectrum
HMI	Human Machine Interface
ICI	Inter-carrier interference
ICS	Industrial Control System
ISI	Inter-symbol interference
IEEE	Institute of Electrical and Electronics Engineers
KiB	Kibibyte
LAD	Ladder Diagram
MIS	Management Information System
OFDM	Orthogonal Frequency Division Multiplexing
OPC	Open Process Control (původně OLE for Process Control)
PLC	Programable Logic Controller, programovatelný logický automat
PyPi	Python Package Index
SCADA	Supervisory Control And Data Acquisition
SCM	Supply Chain Management
STL	Statement List
WSGI	Web Server Gateway Interface

SEZNAM OBRÁZKŮ

Obr. 1. Schéma PLC. [10].....	13
Obr. 2. Základní uspořádání PLC. [10]	14
Obr. 3. Diskrétní, digitální a analogový signál. [10]	15
Obr. 4. Cyklické vykonávání programu. [15].....	15
Obr. 5. Vnitřní uspořádání PLC. [10]	16
Obr. 6. PLC SIMATIC S7-300, CPU. [41]	19
Obr. 7. PLC SIMATIC S7-300, CPU. [42]	20
Obr. 8. PLC SIMATIC S7-1500. [39]	21
Obr. 9. Příklad konfigurace automatizačního systému S7-1500. [39].....	21
Obr. 10. Příklad PAC – SIMATIC IPC227E (Nanobox PC). [34].....	23
Obr. 11. Vrstvy průmyslové sítě. [22]	24
Obr. 12. Síťová topologie. [10].....	25
Obr. 13. Volba nepřekrývajících se frekv.pásem v standardu IEEE 802.11b. [10].....	28
Obr. 14. Jednoduchá topologie systému SCADA. [35].....	31
Obr. 15. Průměrný čas (v hodinách) na vyřešení problému. [29].....	39
Obr. 16. Příklad hardwarové konfigurace PLC řady S7-300 (Simatic Manager).....	41
Obr. 17. Příklad hardwarové konfigurace PLC řady S7-400 (Simatic Manager).....	41
Obr. 18. Vizualizace hardwarové konfigurace CPU 1510SP-1 PN (TIA Portal).....	41
Obr. 19. Zapouzdření protokolů. [40].....	43
Obr. 20. Dialogové okno Module Information – Communication (S7-400).....	44
Obr. 21. Dialogové okno Module Information – Communication (S7-1500).....	45
Obr. 22. Dialogové okno Module Information – Scan Cycle Time.	46
Obr. 23. Vypnutí funkce „Optimized block access“.....	48
Obr. 24. Nastavení úrovně ochrany.	48
Obr. 25. PLC SIMATIC s Raspberry Pi 3, model B.	52
Obr. 26. Srovnání používaných databázových systémů. [33].....	53
Obr. 27. Formulář pro nahrání souboru CSV a volba zdroje dat.....	60
Obr. 28. Ukázka webové prezentace s interaktivním grafem.	62
Obr. 29. Ukázka exportu do formátu PNG.	63
Obr. 30. Ukázka exportu do formátu PNG.	63
Obr. 31. Vykreslený graf s výkonnostní charakteristikou prohlížeče.....	64
Obr. 32. Příklad výpisu funkce cProfile.	65

SEZNAM TABULEK

Tab. 1. Referenční model OSI a jednotlivé funkce vrstev.....	26
Tab. 2. Porovnání architektury TCP/IP s referenčním modelem OSI.....	27
Tab. 3. Přehled nejběžnějších operandů.....	49
Tab. 4. Identifikátory jednotlivých oblastí ve Snap7.....	50
Tab. 5. Přehled použitých databázových systémů.	53

SEZNAM PŘÍLOH

P I Zdrojový kód