TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Porovnání průmyslových komunikačních sběrnic
The Industrial networks comparison

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie

Autor práce: **Jiří Sál**
Vedoucí práce: Ing. Miloš Hernych

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Zadání bakalářské práce

Porovnání průmyslových komunikačních sběrnic

Jméno a příjmení: **Jiří Sál**
Osobní číslo: M15000050
Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie
Zadávající katedra: Ústav mechatroniky a technické informatiky
Akademický rok: **2018/2019**

Zásady pro vypracování:

1. Seznamte se s možnostmi, vlastnostmi a principy moderních průmyslových sběrnic založených na Ethernetu.
2. Navrhněte metodiku pro testování sběrnic a vzájemné porovnání v praktickém nasazení.
3. Navrhněte způsoby propojení testovaných sběrnic s desktopovou aplikací.
4. Navrženou metodiku na vybraných průmyslových sběrnicích implementujte a otestujte.
5. Porovnejte výsledky a zhodnoťte jejich výhody a nevýhody.

Rozsah grafických prací: dle potřeby dokumentace

Rozsah pracovní zprávy: 30–40 stran *Forma zpracování práce:* tištěná/elektronická

Seznam odborné literatury:

[1] https://www.ethercat.org/download/documents/ETG_Brochure_EN.pdf [online]. [cit. 2018-09-12].

[2] PIGAN, Raimond a Mark METTER. Automating with PROFINET: industrial communication based on industrial Ethernet. 2nd rev. and expanded ed. Erlangen: Publicis Pub., 2008. ISBN 9783895782947.

[3] MARSHALL, Perry S. a John S. RINALDI. Industrial Ethernet. 3. Durham: International Society of Automation, 2016. ISBN 1945541040.

Vedoucí práce:	Ing. Miloš Hernych Ústav mechatroniky a technické informatiky
Konzultant práce:	Michal Čermák Cermitech, spol. s r. o.
Datum zadání práce:	10. října 2018
Předpokládaný termín odevzdání:	30. dubna 2019

L. S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci 10. října 2018

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mého bakalářského projektu pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jeho využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Poděkování

Rád bych tímto poděkoval vedoucímu mé bakalářské práce, kterým byl Ing. Miloš Hernych, za průběžnou kontrolu a udávání směru práce, také bych mu rád poděkoval za vřelý přístup a odborné rady.

Poděkování určitě patří i mému konzultantovi panu Michalu Čermákovi za poskytnutí zázemí, sdílení znalostí a poskytování testovacích zařízení.

Abstrakt

Cílem této Bakalářské práce je prozkoumání nabízených možností průmyslové komunikace, tyto možnosti popsat a poté vytvořit jejich porovnání. K dosažení tohoto cíle práce je nutné se seznámit s průmyslovými sběrnicemi založenými na ethernetu. Následně ve vývojovém prostředí CODESYS vytvořit simulace jednoduché výměny dat na vybraných průmyslových sběrnicích, tyto komunikace zachytit a podrobně popsat. Poté navrhnout možnosti propojení desktopové aplikace se zařízeními za použití průmyslových protokolů pomocí vybraných knihoven.

Klíčová slova

Sběrnicemi, ethernetu, CODESYS, simulace, desktopové

Abstract

The objective of this Bachelor thesis is to explore possibilities of industrial comunication, describe these possibilities and create their comparison. To achieve this objective its neccesary to gain knowledge about fieldbuses based on ethernet. Subsequently create simulations of easy data transfer on selected fieldbuses in development environment CODESYS, then capture the communication and describe it. Suggest possibilities of connection between desktop aplication and field device with usage of industrial protocols offered by selected libraries.

Keywords

Fieldbus, ethernet, CODESYS, simulation, desktop

Seznam obrázků

Obrázek 1: ISO/OSI model Zdroj: vlastní zpracování 17
Obrázek 2: Základní prostředí CODESYS Zdroj: vlastní zpracování 40
Obrázek 3: Raspberry Pi 2 Model B 1GB Zdroj: vlastní zpracování 41
Obrázek 4: Instalace CODESYS runtime do Raspberry Pi Zdroj: vlastní zpracování 42
Obrázek 5: Struktura projektu komunikace Modbus TCP Zdroj: vlastní zpracování 43
Obrázek 6: Nastavení ethernetových rozhraní Zdroj: vlastní zpracování 44
Obrázek 7: Nastavení mapování Modbus TCP komunikace Zdroj: vlastní zpracování 44
Obrázek 8: Nastavení komunikačních kanálů Zdroj: vlastní zpracování 44
Obrázek 9: Modbus TCP komunikace ve WireSharku Zdroj: vlastní zpracování 45
Obrázek 10: Struktura projektu komunikace ProfiNet Zdroj: vlastní zpracování 46
Obrázek 11: Nastavení ethernetových rozhraní Zdroj: vlastní zpracování 47
Obrázek 12: Nastavení mapování ProfiNet komunikace Zdroj: vlastní zpracování 47
Obrázek 13: ProfiNet komunikace ve WireSharku Zdroj: vlastní zpracování 47
Obrázek 14: EtherCAT IO-Link Master zařízení od firmy Balluff, s IO-Linkovými vstupy Zdroj: vlastní zpracování 48
Obrázek 15: EtherCAT Coupler of firmy Beckhoff Zdroj: vlastní zpracování 48
Obrázek 16: Struktura projektu EtherCAT komunikace Zdroj: vlastní zpracování 49
Obrázek 17: EtherCAT komunikace ve WireSharku Zdroj: vlastní zpracování 50

Seznam tabulek

Tabulka 1: Schéma hlavičky protokolu IP Zdroj: vlastní zpracování 19
Tabulka 2: Schéma hlavičky protokolu TCP Zdroj: vlastní zpracování 21
Tabulka 3: Schéma hlavičky protokolu UDP Zdroj: vlastní zpracování..... 23
Tabulka 4: Seznam podporovaných registrů v Modbusu Zdroj: vlastní zpracování 31
Tabulka 5: Funkční zprávy podporované Modbusem Zdroj: vlastní zpracování 32
Tabulka 6: Seznam výjimkových zpráv, jež mohou v Modbus komunikaci nastat Zdroj: vlastní zpracování..... 33

Seznam zkratek

APDU	Application Protocol Data Unit
CAN	Controller Area Network
CBA	Component Based Architecture
CIP	Common Industrial Protocol
CN	Controlled Node
COTP	Connection Oriented Transport Protocol
CoDeSys	Controlled Development Systém
HMI	Human Machine Interface
ID	Identifier identifikátor
IO	Input/Output
IP	Internet Protokol
IRT	Isochronous Real Time
ISO	International Organization for Standardization
LAN	Local Area Network
LED	Light-Emitting Diode
MAC	Media Access Control fyzická adresa
MN	Managing Node
OLE	Object Linking Embedding
OPC	OLE for Process Control
OS	Operační systém
OSI	Open Systems Interconnection
PC	Personal Computer osobní počítač
PDU	Protocol Data Unit
PLC	Programmable Logic Controller
PLC	Programmable Logic Controller
PPDU	Presentation Protocol Data Unit
ProfiBus	Process Field Bus
ProfiBus DP	Process Field Bus Decentralized Periphery
ProfiBus PA	Process Field Bus Process Automation
RTU	Remote Terminal Unit
SD	Secure digital
SOEM	Simple Open EtherCAT Master
SPDU	Session Protocol Data Unit
SSH	Secure Shell
TCP	Transmission Control Protocol
TPDU	Transport Protocol Data Unit
TPKT	Transport Packet
TTL	Time To Live
UDP	User Datagram Protocol
WKC	Working Counter

Obsah

1. Úvod ... 12
2. Průmyslové sběrnice ... 13
2.1 Průmyslové sběrnice ... 13
2.1.1 CAN bus ... 13
2.1.2 ProfiBus ... 14
2.1.3 Modbus ... 15
2.2 Průmyslový ethernet ... 15
2.2.1 Ethernet (IEEE 802.3) ... 16
2.2.2 Vrstvy ISO/OSI ... 17
2.2.3 Důležité protokoly ... 18
2.3 ProfiNet ... 24
2.3.1 ProfiNet IO ... 25
2.4 EtherCAT ... 25
2.4.1 Diagnostika ... 26
2.5 Siemens S7 Communication ... 28
2.5.1 S7 PDU ... 29
2.6 Modbus TCP/IP ... 30
2.6.1 Modbusové funkce a registry ... 31
2.7 Ethernet PowerLink ... 34
2.8 EtherNet/IP ... 36
2.9 Další způsoby komunikací (OPC, I/O Link) ... 37
2.9.1 OPC ... 37
2.9.2 Příklady architektury OPC klient-server ... 38
2.9.3 IO-Link ... 38
3. Použité technologie ... 40
3.1 CODESYS ... 40
3.2 Raspberry Pi ... 41
3.2.1 Připojení Raspberry Pi s CODESYS ... 41
3.3 WireShark ... 42
3.3.1 WinPcap ... 42
4. Simulace ... 43
4.1 Modbus TCP ... 43

4.1.1 Struktura projektu 43
4.1.2 Nastavení Modbus TCP simulace 44
4.1.3 Komunikace ve WireShark 45
4.2 ProfiNet 45
4.2.1 Struktura projektu 46
4.2.2 Nastavení ProfiNet simulace 46
4.2.3 Komunikace ve WireShark 47
4.3 EtherCAT 48
4.3.1 Zařízení 48
4.3.2 Struktura projektu 49
4.3.3 Komunikace ve WireShark 50
5. Porovnání průmyslových sběrnic 51
6. Způsoby propojení desktopových aplikací 53
6.1 Python-snap7 53
6.2 EasyModbusTCP/UDP/RTU 53
6.3 Simple Open EtherCAT Master Library 53
7. Závěr 54
Citovaná literatura 55

1. Úvod

Průmyslové sběrnice jsou nedílnou součástí průmyslové robotiky a automatizace. Díky možným složitým fyzickým podmínkám musí být mechanicky odolné vůči samotnému fyzickému poškození, ale také vůči pravděpodobnému elektromagnetickému rušení, které se může a nejpravděpodobněji bude, na místě jejich působení vyskytovat. V dnešní době již nestačí pouze odolnost, ale sběrnice musí dosahovat co nejvyšších přenosových rychlostí, kvůli vzrůstajícím potřebám řízení v reálném čase.

V této práci bychom měli porozumět důvodu vzniku průmyslových sběrnic, jejich historii, vývoji a hlavním využití jejich služeb. Dále budou zmíněny protokoly, které různé průmyslové sběrnice využívají.

V dnešní době existuje několik průmyslových sběrnic od různých výrobců s různými způsoby komunikace a v této bakalářské práci bychom se měli seznámit s rozdíly mezi jednotlivými průmyslovými sběrnicemi založenými na Ethernetu, hlavní jsou rozdíly ve způsobu komunikace, dalším velkým kritériem průmyslových sběrnic je poskytování komunikace v reálném čase.

Ukážeme si simulaci komunikace vybraných průmyslových sběrnic což bude prováděno ve vývojovém prostředí CODESYS, kde se podíváme na projekty, ve kterých bude probíhat simulace, kde spolu budou komunikovat vždy dvě zařízení po zvolené průmyslové sběrnici.

Poté budou navrženy možnosti komunikace zařízeních s desktopovou aplikací pomocí průmyslových sběrnic za použití vybraných existujících knihoven.

2. Průmyslové sběrnice

2.1 Průmyslové sběrnice

Průmyslové sběrnice představují již nenahraditelnou součást průmyslových aplikací, může se jednat o jednoduché získávání dat z průmyslových zařízení nebo řízení výroby. K tomuto účelu je zapotřebí nejméně jednoho řídicího zařízení a senzoru.

Vznikly za účelem zjednodušení a ucelení komunikace jednotlivých zařízení s jejich řídicím prvkem. Komunikace probíhá za použití průmyslových komunikačních protokolů, které plní požadavky stanovené výrobními procesy. Průmyslový komunikační protokol je sada pravidel, která určuje parametry přenosu dat mezi zařízeními.

V ranných fázích zavádění technologie průmyslových sběrnic, byla automatizace omezena na lokální řízení určitých strojů nebo výrobních linek, to znamená oddělení automatizačních systémů, které mezi sebou nemohou sdílet informace pro optimalizaci výrobních procesů. Prvním krokem k ucelené komunikaci bylo propojení těchto oddělených automatizačních systémů. Dalším krokem bylo propojení již spojených automatizačních systémů s řídicím zařízením, které by bylo schopné pracovat s poskytovanými daty, kontrolovat stavy k němu připojených zařízení a poskytovat náhled na získávaná data.

Průmyslových protokolů existuje několik a každý poskytuje svou sadu pravidel k realizaci komunikace. [1] [2]

2.1.1 CAN bus

CAN je komunikační protokol vyvinutý společností Bosch, který definuje, jak jsou data doručována z jednoho zařízení do druhého a využívá se nejčastěji pro vnitřní komunikační sít senzorů, převážně je užívaný v automobilovém průmyslu, v něm se využívá dodnes ke konstruování řídicí sítě v automobilech, první automobil, který měl tuto technologii nainstalovanou, bylo v roce 1986 BMW 850, díky této sběrnici se snížila délka použitých kabelů o 2 km a celková váha vozu o 50 kg. Dnes je možné nahradit sběrnici CAN v automobilech optickými kabely.

Komunikace pomocí tohoto protokolu je asynchronní a sériová, její parametry jsou definované normou ISO 1189 [3], maximální teoretická rychlost přenosu dat je 1 Mb/s. Zprávy jsou doručovány na základě priority, která je určena v typu zprávy identifikačním číslem, identifikátor určuje nejen prioritu, ale i význam zprávy. Čím má zpráva menší identifikátor, její priorita je vyšší. Zpráva je obal, který je složený ze signálů, v jedné zprávě musí být minimálně 1 a maximálně 64 signálů. Zpráva může být přijata zároveň několika zařízeními. Vychází z referenčního modelu ISO/OSI a užívá ke komunikaci jeho dvou nejnižších vrstev fyzickou a linkovou. Protokol samotný nabízí možnost detekce přenosových chyb vzniklých okolními elektromagnetickými poli.

Podporuje takzvanou plug-and-play funkci, která značí, že je možné v průběhu komunikace přidávat uzly. Uzly jsou schopné se po zapojení začlenit do sítě. Každé zařízení obsahuje svůj mikrokontroler, jenž je schopen výměny dat s ostatními zařízeními. [4] [5]

2.1.2 ProfiBus

ProfiBus je průmyslová sběrnice vyvinuta v roce 1989. Je užívaná k řízení výroby a automatizaci výrobních linek, používá metodu master-slave pro komunikaci mezi aktivním řídícím zařízením a jemu přiděleným zařízením. K řízení přístupu na sběrnici používá metodu token ring. Rychlost sítě je omezena její maximální délkou segmentu. Existují dvě nejrozšířenější verze této sběrnice ProfiBus DP a ProfiBus PA, tyto verze se používají i dnes. Z modelu ISO/OSI staví na třech vrstvách, a to na vrstvě aplikační, linkové a fyzické.

ProfiBus s decentralizovanými periferiemi je nejrozšířenější verzí ProfBusu, je určená k rychlé cyklické komunikaci typu master-slave. Podporuje dvě přenosová média kroucenou dvoulinku, nebo optické vlákno. Je vybaven funkcemi pro diagnostiku a monitorování stavu sítě z hlediska bezpečnosti a spolehlivosti. Na jeden segment sítě může být připojeno až 32 aktivních nebo pasivních zařízení. Pro připojení dalších zařízení je zapotřebí užití opakovače, který rozšíří síť o další segment a zvýší maximální počet připojitelných zařízení až na 127.

ProfiBus pro procesní automatizaci rozšiřuje ProfiBus DP a je určen pro řízení pomalých procesů převážně v prostředí rizikovém ke vzplanutí, splňuje totiž požadavky jiskrové bezpečnosti. ProfiBus PA je varianta DP, realizovaná na proudové smyčce dle normy IEC 61158 [6]. Používá se v aplikacích, ve kterých je třeba přenášet data na velkou vzdálenost a kde není kladen důraz na rychlost. [7]

2.1.3 Modbus

Modbus je master-slave orientovaný otevřený protokol publikovaný v roce 1979 firmou Modicon a popisuje strukturu sériové komunikace mezi dvěma inteligentními zařízeními. Na jedno master zařízení připadá maximálně 247 slave zařízení s unikátními adresami.

Modbus je často používán k propojení kontrolního počítače se vzdálenou terminálovou jednotkou v systémech získávání dat nebo kontroly výrobního procesu. Modbus nabízí tři verze z nichž dvě verze existují pro použití na sériových linkách (Modbus RTU, Modbus ASCII) a jedna ho rozšiřuje na Ethernet (Modbus TCP/IP). [8] [9]

2.2 Průmyslový ethernet

Pojmem Průmyslový ethernet označujeme průmyslovou sběrnici, která v nějakém směru využívá standartní ethernet. Na rozdíl od běžného ethernetu je na průmyslový ethernet kladeno několik požadavků navíc a to:

- Včasné a současné plnění požadavků jednotlivých komponent podílejících se na komunikaci podle předem zadaných priorit.
- Včasné a současné reagování na výstražná hlášení.
- Minimalizaci kolísání doby odezvy.
- Stabilita hardwarových komponent, ochrana před výpadkem zařízení.
- Stabilita softwarových komponent, stabilní operační systém účastníků sítě a kvalitní síťový software.
- Odolnost proti mechanickému poškození (vibrace, nárazy).

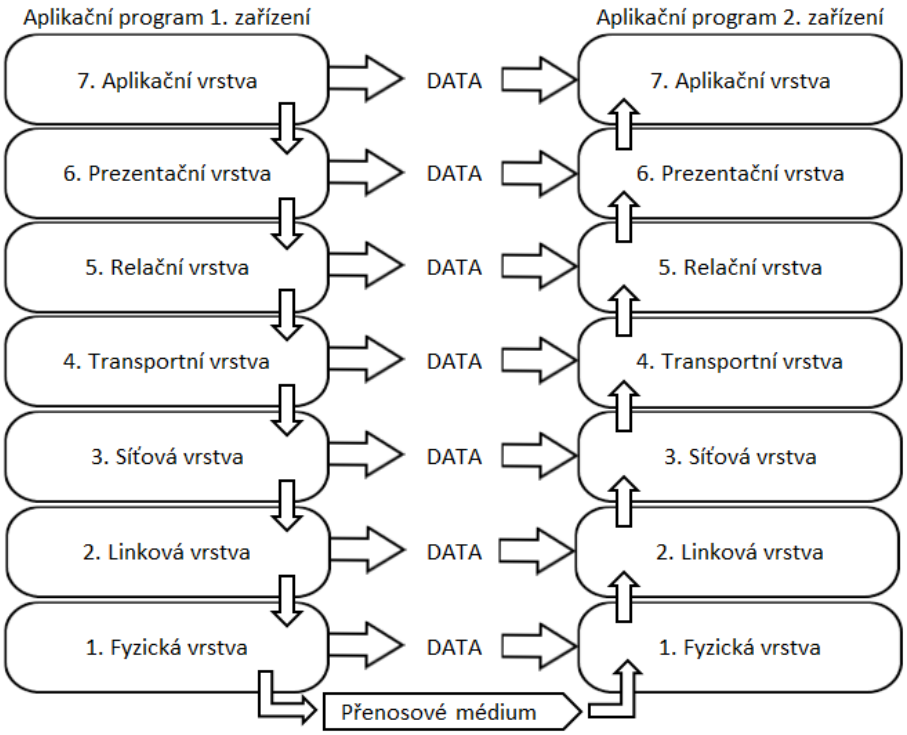
- Odolnost vůči průmyslovému znečištění (olej, chemikálie, vlhkost, elektromagnetické rušení). [10] [11]

Průmyslový ethernet byl standardizován pod normou IEC 61158 [6], jedná se o mezinárodní standard, který definuje způsoby komunikace po průmyslových sběrnicích.

2.2.1 Ethernet (IEEE 802.3)

Abychom mohli porozumět průmyslovému Ethernetu, musíme nejdříve pochopit samotný ethernet. Ethernet je v dnešní době nejrozšířenější komunikační standard především pokud se jedná o jednoduché lokální sítě (LAN). Je důležité rozlišovat Ethernet jako označení komunikační sítě a Ethernet jako název označující souhrn komunikačních protokolů, určujících pravidla komunikace po síti.

Na začátku 80. let dvacátého století vznikl za účelem vymezení pravidel propojování hardwarově nebo softwarově různorodých zařízení tzv. referenční model ISO/OSI, byl navržen mezinárodní standardizační organizací ISO. Tento model obsahuje sedm vrstev, z nichž každá vrstva logicky komunikuje se stejnou vrstvou dalšího zařízení, fyzicky však komunikuje s nižší vrstvou na svém zařízení, dochází k tzv. zapouzdřování dat, ty jsou pak příslušnými vrstvami rozbalovány. Komunikaci mezi zařízeními začíná aplikační vrstvou odesílajícího zařízení, ta obdrží vstupně-výstupní data, která poté zasílá napříč vrstvami až na přenosové médium, adresátova fyzická vrstva tato data příjme a předává vyšším vrstvám, až se dostanou do aplikační vrstvy, zde jsou zpracována a použita dle hlavičky aplikační vrstvy odesílatele.



Obrázek 1: ISO/OSI model
Zdroj: vlastní zpracování

Na tomto obrázku vidíme schéma referenčního modelu ISO/OSI, je na něm znázorněno, že data jednotlivých vrstev komunikují se stejnou vrstvou na dalším zařízení, to je prováděno přidáváním vrstvových hlaviček, které uchovávají informace o používaných protokolech a způsobu komunikace. [12]

2.2.2 Vrstvy ISO/OSI

Fyzická vrstva

Nejnižší vrstva referenčního modelu, která specifikuje samotnou fyzickou komunikaci. Jejím úkolem je inicializace, udržování a ukončování fyzického spojení mezi koncovými systémy. V této vrstvě se definují všechny elektrické a fyzikální parametry zařízení. Stanovuje způsob přenosu bitů. Přenosovou jednotkou je bit.

Linková vrstva

Linková vrstva poskytuje spojení mezi dvěma sousedními uzly sítě a zodpovídá za organizaci dat do rámců a zajištění hop-by-hop doručení dat. Na této vrstvě aktivně operují rozdělovače. Přenosovou jednotkou je rámec.

Síťová vrstva

Cílem síťové vrstvy je sestavení a doručení paketů odesílatele k příjemci, doručení paketu může někdy znamenat cestu přes několik nezávislých sítí, tato funkce se nazývá směrování. Na této vrstvě aktivně operují směrovače a používá se zde pro nás důležitý protokol IP, směrovače umožňují zasílání paketů mezi sítěmi, které používají jiný protokol linkové vrstvy. Jednou z funkcí této vrstvy je, že překládá logickou adresu na fyzickou. Přenosovou jednotkou je paket.

Transportní vrstva

Transportní vrstva zajištuje spolehlivé doručení zprávy sítí mezi koncovými uzly, poskytuje mechanismy na detekci chyb a kontroluje proud dat. V transportní vrstvě se používají pro nás dva důležité protokoly TCP a UDP. Obdrží data z vyšší vrstvy a rozdělí je do menších segmentů, které předává síťové vrstvě. V opačném směru tyto menší segmenty znovu spojuje. Přenosovou jednotkou je TPDU.

Relační vrstva

Funkce této vrstvy je taková, že navazuje, udržuje a spravuje spojení mezi aplikacemi, poskytuje služby jako autentikace, autorizace. Přenosovou jednotkou je SPDU.

Prezentační vrstva

Prezentační vrstva transformuje data do takové podoby, které aplikační vrstva rozumí a bude s nimi moci pracovat. Přenosovou jednotkou je PPDU.

Aplikační vrstva

Nejvyšší vrstva referenčního modelu, s touto vrstvou pracuje koncový uživatel, ať se jedná o programátora nebo obyčejného uživatele. V této vrstvě se definuje požadované spojení. Přenosovou jednotkou je APDU. [13] [14]

2.2.3 Důležité protokoly

V následující kapitole budou popsány jednotlivé důležité protokoly využívané ke komunikaci v některých průmyslových sběrnicích.

Internet Protocol

Jedná se nejpoužívanější protokol pro komunikaci v počítačových sítích. Pracuje na síťové vrstvě referenčního modelu ISO/OSI. Tento protokol přiřazuje k rozhraní jeho logickou adresu. V současné době se nejvíce využívá IPv4 verze tohoto protokolu,

která pracuje s 32bitovými adresami, což znamená, že poskytuje 4,294,967,296 celkových logických adres. Z důvodu velké popularity tohoto protokolu a celkového světového rozšíření počítačových sítí tyto adresy pomalu docházejí, proto vznikla novější verze protokolu IP, která se jmenuje IPv6 a pracuje s 128bitovými adresami, celkový počet adres v této verzi dalece přesahuje jeho původní verzi (7,9x10^28krát). Protokolové číslo pro IPv4 je 4. [15]

IP protokol pracuje s IP paketem, ten je vytvořen přidáním IP hlavičky do každého příchozího segmentu data před odesláním do nižší vrstvy. Hlavička IP paketu se skládá z několika informací o paketu, mezi ně patří cílová a výchozí IP adresa, TTL, informace o použitém protokolu, celková délka a další. Celkově má IPv4 hlavička 14 parametrů z nichž jeden je volitelný. [16]

Tabulka 1: Schéma hlavičky protokolu IP
Zdroj: vlastní zpracování

<table>
<tr><td>Verze (4b)</td><td>Délka hlavičky (4b)</td><td>Typ služby (8b)</td><td colspan="2">Celková délka (16b)</td></tr>
<tr><td colspan="3">Identifikace (16b)</td><td>Příznaky (3b)</td><td>Offset fragmentu (13b)</td></tr>
<tr><td colspan="2">TTL (8b)</td><td>Protokol (8b)</td><td colspan="2">Hlavičkový kontrolní součet (16b)</td></tr>
<tr><td colspan="5">Zdrojová IP adresa (32b)</td></tr>
<tr><td colspan="5">Cílová IP adresa (32b)</td></tr>
<tr><td colspan="5">Možnosti (0 – 32b)</td></tr>
</table>

Na schématu jsou k vidění všechny parametry použité k popsání komunikačního protokolu IP, funkce jednotlivých polí jsou tyto:

- Verze – identifikuje verzi použitého IP protokolu, pro IPv4 má toto pole hodnotu 4.
- Délka hlavičky – specifikuje celkovou délku hlavičky. Minimální hodnota je 20B a maximální je 60B.

- Typ služby – určuje, jak nakládat se zaslaným datagramem.
- Celková délka – délka celého paketu (hlavička + data), minimální délka je 20B a maximální 65 565B.
- Identifikace – je přidělena paketům jasná identifikace, kvůli možné fragmentaci při průchodu sítí, určuje skupině paketů příslušící jedné zprávě stejný identifikátor.
- Příznaky – slouží k identifikaci a kontrole fragmentace, první příznakový bit je vždy nulový, druhý je bit, který určuje, zdali je možné tento paket fragmentovat a třetí určuje, jestli se nejedná o poslední paket ze série fragmentovaných paketů. Poslední příznakový bit má hodnotu nastavenou na logickou jedna pro všechny pakety kromě posledního.
- Offset fragmentu – toto pole umožňuje cílovému zařízení sestavení fragmentované sekvence do původního paketu. Pro první fragment je vždy nulový a maximální možný offset je 8191.
- TTL – určuje životnost datagramu, jedná se o opatření proti zacyklení, každé zařízení pracující se síťovou vrstvou dekrementuje TTL o jedna, pokud se TTL zmenší až na hodnotu 0, aniž by se datagram dostal do cíle, je zahozen.
- Protokol – definuje protokol vyšší vrstvy.
- Hlavičkový kontrolní součet – slouží ke kontrole chyb hlavičky, pokud paket dorazí na směrovač a směrovač vypočítá jiný kontrolní součet, než je v hlavičce uveden, je tento paket zahozen.
- Zdrojová IP adresa – IP adresa vysílajícího paket.
- Cílová IP adresa – IP adresa zařízení, které by mělo paket přijmout.
- Možnosti – pole pro rozšiřující možnosti, užívané pro testy sítě nebo její bezpečnosti. [17]

Transmission Control Protocol

Je protokolem transportní vrstvy a je jedním z hlavních protokolů ethernetové komunikace, je často nazýván TCP/IP protokolem, kvůli jeho úzké spolupráci s IP protokolem. TCP poskytuje spolehlivou, organizovanou a kontrolovanou datovou výměnu mezi běžícími zařízeními komunikujícími přes IP, pakety se cestou sítí mohou ztratit, nebo změnit pořadí, v takovém případě TCP žádá o přeposlání zprávy nebo je seřadí do jejich původní podoby, aproto se používá pro zasílání takových dat, které musí spolehlivě dorazit k příjemci v určitém pořadí. Spoléhají se na něj internetové aplikace jako World Wide Web, vzdálená správa a jiné. Protokolové číslo pro TCP je 6. [18]

Tabulka 2: Schéma hlavičky protokolu TCP
Zdroj: vlastní zpracování

<table>
<tr><td colspan="3">Zdrojový port (16b)</td><td>Cílový port (16b)</td></tr>
<tr><td colspan="4">Číslo sekvence (32b)</td></tr>
<tr><td colspan="4">Potvrzovací číslo (32b)</td></tr>
<tr><td>Délka hlavičky (4b)</td><td>Rezer-vováno (3b)</td><td>Kontrolní příznaky (9b)</td><td>Velikost okénka (16b)</td></tr>
<tr><td colspan="3">Kontrolní součet (16b)</td><td>Urgent pointer</td></tr>
<tr><td colspan="4">Možnosti (0 – 32b)</td></tr>
</table>

Na schématu je znázorněna hlavička TCP protokolu, jednotlivé parametry plní tyto úkony:

- Zdrojový port – určuje port aplikace, ke kterému paket náleží.
- Cílový port – určuje port aplikace, do které má paket putovat.
- Číslo sekvence – se používá k segmentaci zprávy u odesílatele a její sestavení u příjemce. Pomáhá obou stranám hlídat, kolik dat bylo již přeneseno a pokud byla data obdržena ve špatném pořadí, slouží k jejich řádnému sestavení.
- Potvrzovací číslo – toto pole obsahuje číslo sekvence následujícího segmentu, který je očekáván, na začátku přenosu není nastaveno.

- Délka hlavičky – indikuje celkovou délku hlavičky. Maximální délka hlavičky je 60B a minimální 20B.
- Rezervováno – nastaveno na nula.
- Kontrolní příznaky – obsahuje 9 jednobitových příznaků:
    - NS – je příznak, který pomáhá chránit odesílaný paket, jedná se o experimentální příznak.
    - CWR – značí odesílateli, že obdržel paket s nastaveným ECE příznakem.
    - ECE – má dva významy dle SYN příznaku.
    - URG – indikuje že pole urgent pointer obsahuje validní data
    - ACK – indikuje že pole potvrzovací číslo obsahuje validní data.
    - PSH – znamená důležitý přenos, který by se měl do cílové aplikace dostat přednostně.
    - RST – pokud je nastaven, dojde k restartování přenosu.
    - SYN – synchronizace sekvenčních čísel, měl by být nastaven pouze v prvním datagramu z vysílajícího a přijímacího zařízení.
    - FIN – indikuje poslední datagram odesílatele.
- Velikost okénka – určuje, jak velká data mohou být zaslána před potvrzením jejich obdržení. Pokud je okénko nastaveno na zbytečně nízkou hodnotu, dochází k nepotřebnému zpomalení přenosu dat, zatímco pokud je nastaveno na vysokou hodnotu, může dojít k zahlcení sítě, nebo příjemce nebude schopen data včas zpracovávat.
- Kontrolní součet – je generován odesílatelem jako technika, která pomáhá příjemci detekovat poškozený přenos.
- Urgent pointer – je většinou nastaven na nulu a ignorován, avšak s nastaveným příznakem URG odkazuje na část dat, která jsou potřeba vyřídit prioritně.
- Možnosti – pole pro rozšiřující možnosti. [19] [20] [21] [22]

User Datagram Protocol

Je druhým hlavním protokolem transportní vrstvy, který slouží pro opačný účel než TCP, je používán pro zprávy, které jsou třeba doručit co nejrychleji. Na rozdíl od TCP nenavazuje spojení a je nespolehlivý, ale tyto nevýhody jsou vynahrazeny dosahovanou rychlostí, je využit například pro přenos videa, kde nevadí, když je pár paketů cestou ztraceno. Protokolové číslo pro UDP je 17. [23]

Tabulka 3: Schéma hlavičky protokolu UDP
Zdroj: vlastní zpracování

Zdrojový port (16b)	Cílový port (16b)
Délka (16b)	Kontrolní součet (16b)

Jak je ze schématu patrné, hlavička UDP protokolu je značně jednodušší než hlavička TCP protokolu, jednotlivé parametry mají tyto významy:

- Zdrojový port – určuje port aplikace, ke kterému paket náleží.
- Cílový port – určuje port aplikace, do které má paket putovat.
- Délka – reprezentuje celkovou délku každého datagramu.
- Kontrolní součet – podobně jako u TCP slouží k detekci chybných dat.

CIP

Je otevřený průmyslový protokol, využívaný pro automatizační aplikace a je použit například v EtherNet/IP protokolu. Obsahuje kompletní seznam služeb a zpráv pro kolekci výrobních automatizačních aplikací (řízení, bezpečnost, pohyb a jiné). Povoluje uživateli integrovat tyto výrobní aplikace na ethernetové sítě a internet. Je nezávislý na médiu, a proto je velmi podporovaný. Existuje několik protokolů využívající CIP jsou to EtherNet/IP, DeviceNet, ControlNet a CompoNet, každý z těchto způsobů definuje svůj specifický objekt, EtherNet/IP definuje dva objekty, jeden pro TCP/IP a druhý pro Ethernet.

CIP je striktně objektově orientovaný a každý uzel je modelován jako množina objektů v definovaném tvaru. Každý objekt representuje určitý element uzlu jako příslušnost k jistému typu podporovaných zařízení, všechna zařízení stejného typu mají

stejné atributy a chovají se stejně. Jeden uzel může obsahovat více než jeden objekt stejného typu. [24]

2.3 ProfiNet

Dostáváme se k první průmyslové sběrnici ProfiNet. ProfiNet je otevřený, na výrobci nezávislý standard pro průmyslový ethernet. Byl vyvinut organizací Profibus Nutzer/user Organization ve spolupráci s firmou Siemens a je založen na průmyslové sběrnici ProfiBus, kterou rozšiřuje na ethernet. Podporuje tři topologie linii, větev a strom, i když je podporována topologie do linie, nedoporučuje se ji používat.

Pracuje na čtyřech vrstvách referenčního modelu ISO/OSI, a to na vrstvě aplikační, transportní, síťové a fyzické. Nemusí však používat všechny čtyři vrstvy, to záleží na použití komunikačního mechanizmu, ProfiNet používá tři, normální způsob komunikace, který používá pro datové přenosy protokoly TCP/IP a UDP/IP. Tento způsob komunikace se užívá pro zprávy, které nejsou potřeba doručit co nejrychleji, jako parametrizace a nastavení komunikace. Tomuto komunikačnímu způsobu se také říká CBA.

Dalším způsobem komunikace je real-time neboli komunikace v reálném čase, který vynechá třetí a čtvrtou vrstvu referenčního modelu ISO/OSI, tento způsob komunikace je používán pro takový typ dat, jenž je kritický pro výrobu a vyžaduje co nejmenší odezvu, je cyklický a spouštěný lokálními časovači. ProfiNet nabízí speciální komunikační kanál, který je určen přímo tomuto typu komunikace.

Posledním způsobem komunikace je takzvaný izochronní reálný čas, zkráceně IRT. Tato komunikace je používána na zvláštní aplikace vyžadující velice nízkou odezvu. Při použití IRT je možné snížit časový cyklus sběrnice až na 1ms, přičemž omezí kolísání odezvy (jiter) na 1μs, což pro tento komunikační typ jasně definuje determinismus. ProfiNet nabízí dva komunikační kanály, které mohou bez kolizí existovat vedle sebe, a to právě kanál pro IRT a kanál pro další komunikaci, který může využívat jeden ze dvou předchozích komunikačních možností. [25]

2.3.1 ProfiNet IO

ProfiNet IO je standard pro ProfiNet, který se stará o komunikaci s periferiemi a přímo definuje komunikaci s připojenými periferními zařízeními. Definuje celý způsob výměny dat mezi kontrolerem a zařízeními. ProfiNet IO používá provider-customer model výměny dat namísto master-slave, který je hojně používaný mezi ostaními průmyslovými sběrnicemi založenými na ethernetu.

V ProfiNet IO, rozlišujeme tři typy zařízení, IO Controller, IO Device a IO Supervisor. IO Controller zpravidla chápeme jako PLC, které řídí IO Device, jenž jsou připojené aktivní prvky jako senzory. IO Supervisor je pak aplikace počítače nebo HMI, která slouží pro kontrolu připojení, správné funkcionality zařízeních nebo pro vizualizaci dat. [26]

2.4 EtherCAT

Další průmyslovou sběrnicí založenou na Ethernetu je EtherCAT, původně byl vyvinut firmou Backhoff, která její vývoj později svěřila do rukou EtherCAT Technology Group spravující EtherCAT dodnes. Jedná se o open-source protokol a jeho komunikace je efektivní a přímá. Má sice omezený počet připojených zařízení pro jeden segment sítě na 35 535 zařízení, avšak není omezeno kolik segmentů v síti existuje. Není designovaný pro standartní TCP/IP pakety a zaměřuje se převážně na komunikaci v reálném čase. Aplikační vrstva EtherCATu je inspirovaná z modelu, který využívá sběrnice CANOpen. EtherCAT využívá princip komunikace master-slave, kde master je typicky řídicí systém posílající rámce s pokyny přiřazeným podřízeným zařízením a slave je řízená jednotka přiřazená k právě jednomu master zařízení, síť EtherCATu může používat různé způsoby zapojení (topologii). Je schopen zpracovat 1000 vstupně-výstupních zařízení za mikrosekundu nebo 100 os (servopohonů, enkodérů, …) za 125 mikrosekund.

Jedna z odlišností EtherCATu od ostatních ethernetových průmyslových sběrnic je ve způsobu komunikace, ta probíhá tak, že master pošle zprávu do celého segmentu jeho sítě, která obsahuje data pro všechny adresované, jemu přiřazené podřízené jednotky. Data jsou poté odebírána při jejich cestě zařízením, tomuto způsobu se říká „on

the fly“, tento způsob komunikace je dobře předvídatelný, což z EtherCATu dělá velice deterministickou sběrnici. Díky „on the fly“ způsobu komunikace je tvořena zmiňovaná rozličnost použitých topologií, dá se kupříkladu udělat takovou topologii, která bude v kritickém segmentu sítě cyklená, tudíž při přerušení jednoho média, bude existovat náhradní cesta do adresovaného zařízení.

EtherCAT nabízí dva způsoby výměny dat cyklický a acyklický, přičemž cyklický je základní způsob komunikace mezi master a slave zařízeními, tento způsob poskytuje výborné rea-time vlastnosti s rozšířenou možností diagnostiky. Acyklický způsob podporuje různé protokoly podporující IP tyto protokoly zpravidla fungují k diagnostice nebo konfiguraci zařízení, za použití *Automation Device Specification over EtherCAT* protokolu, jenž je specifický protokol vyvinutý právě za účelem acyklické komunikace bez narušení cyklické. Po EtherCATu podporované protokoly jsou například CANopen over EtherCAT, Servodrive profiles over EtherCAT, File Access over EtherCAT. [27] [28]

Při komunikaci může nastat několik chyb komunikace, zapříčiněných buď fyzickou (hardwarovou) nebo aplikační (softwarovou) chybou. Jako hardwarovou chybu chápeme přerušené přenosové médium nebo změnu topologie, což způsobí, že data od master zařízení nenajdou cestu ke svému adresátu, další možnou hardwarovou chybou je, že všechny rámce dorazí do adresovaných zařízení, ale bitová sekvence je poškozena (corrupted). Pojmem softwarová chyba rozumíme chybu aplikace, tu může způsobovat, když parametry zaslané master zařízením během start-up fáze (konfigurace), jsou špatné nebo nesplňují očekávání slave zařízení (špatná velikost dat, špatná konfigurace, špatný cycle time), nebo když slave zařízení, jež do určitého času běžel bez chyby (error-free), začne bezdůvodně způsobovat chyby, to může být zapříčiněno ztrátou synchronizace nebo vypršení platnosti dozoru. (watchdog). [29]

2.4.1 Diagnostika

Diagnostika se dělí na hardwarovou a softwarovou a cyklickou a acyklickou. Cyklická diagnostika probíhá v každém PLC cyklu, hlavním cyklickým diagnostickým prostředkem je working counter, kterým je zakončen každý EtherCAT datagram, ten je inkrementován při každém průchodu slave zařízení, pokud se datagram vrátí do master

zařízení s invalidním working counterem, jsou data nesena tímto datagramem zahozena. Informace o working counteru mohou být master zařízením zaslána do kontrolní stanice (PC, HMI). Acyklická diagnostika probíhá mimo PLC cyklus, obvykle se jedná o kontrolní aplikaci.

Hardwarová diagnostika je základní poskytovanou diagnostikou, a jak vychází z názvu je na fyzické úrovni, zpravidla se jedná o chybové čítače poskytované slave jednotkami, které jsou přístupné na paměťových adresách (memory adresses).

1) Master Lost Frame Counter – je čítač ztracených rámců na straně řídící jednotky, rámec je považován za „ztracený“, pokud se vůbec nevrátí z průchodu sítí nebo se vrátí poškozený, a tudíž je informace, kterou nese považována za zbytečnou.
2) Hardware Error Counters – je skupina čítačů, které hlídají chyby na samotném přenosovém médiu.
    a. Link Lost Counter – tento volitelný čítač se přičte pokaždé, co dojde k fyzickému přerušení spojení mezi zařízeními, mezi pravděpodobné příčiny přerušení spojení patří externí elektromagnetické rušení, poškození přenosového média, přerušení dodávky elektřiny nebo reset zařízení.
    b. Link Activity LEDs – není sice čítač, ale slouží k diagnostice hardwarových chyb, každé EtherCAT slave zařízení má povinně LED diodu, která signalizuje fungující komunikaci pro každé síťové rozhraní.
    c. Invalid Frame Counter – tento čítač je povinný a je tedy aktivní nehledě na nastavení komunikace, inkrementuje se při vyskytnutí chyby signalizace a indikuje chybné rámce. Mezi možné příčiny patří elektromagnetické rušení nebo poškozená zařízení.
        i. RX Error Counter – označuje individuální chybný symbol a může nastat uvnitř i vně rámců.
        ii. CRC Error Counter – označuje rámce, jejichž bitová sekvence byla poškozena a vyskytuje se jen uvnitř rámců.

Softwarovou diagnostikou je rozuměna kontrola pomocí chybného stavu zařízení. Každé slave zařízení v EtherCAT síti je řízené EtherCAT stavovým zařízením (master) a hlásí svůj stav (state) nebo příznakovou chybu (flag error) ve stavu zařízení „state machine“, ke kterému se přistupuje na registru Status_register 0x0130. Pokud chce master novou informaci o stavu zařízení, zašle mu zprávu se změnou AL_status registru 0x0120 ve slave zařízení, takováto změna stavu je možná, jen pokud se jedná o error. Stavy slave zařízeních mohou být tyto:

- Init – v tomto stavu není možné se zařízením komunikovat ani cyklicky ani acyklicky
- PreOP – v tomto stavu je možné se zařízením komunikovat acyklicky nikoliv však cyklicky
- SafeOP – oba komunikační způsoby jsou povoleny, avšak cyklické výstupy zůstávají v předdefinovaném stavu
- OP – oba komunikační způsoby povoleny bez omezení
- Boot – volitelný stav pro aktualizaci firmwaru, pouze acyklický přenos [30]

2.5 Siemens S7 Communication

S7 proprietární komunikační protokol je vyvinutý firmou Siemens, slouží k propojení řídicího zařízení sítě (PLC) a kontrolního zařízení (PC). Většinou využívá pro průmyslové sběrnice celkem rozšířený master-slave způsob řízení, v některých případech tento způsob rozšiřuje na klient-server. V tomto případě slouží jako slave jednotka PLC a master PC, existuje však několik výjimek, kdy PLC slouží jako master. S7 protokol je takzvaně function-command orientovaný, to znamená, že komunikace je složena z řádného S7 požadavku a řádné S7 odpovědi, i pro tuto skutečnost existuje několik výjimek. Počet paralelních přenosů a maximální délka PDU je určena při konfiguraci připojení (smluveno při nastavování komunikačních parametrů).

PC (master-klient) posílá S7 požadavky na podřízené zařízení. V této síti existuje možnost peer-to-peer propojení, které začíná vysláním funkční zprávy BLOCK SEND ze strany vysílače a odpovědí o potvrzení tohoto přenosu funkcí BLOCK RECEIVE.

2.5.1 S7 PDU

S7 protokol implementuje službu TCP/IP transportní vrstvy referenčního modelu ISO/OSI, je zabalen do TPKT a ISO-COTP protokolů, což slouží k přenosu S7 PDU pomocí TCP. Skládá se ze tří částí header, parameters a data.

- Header – hlavička obsahuje důležité informace o přenesených datech, a to délku přenášených dat, PDU odkazy a typ zasílané zprávy. Je dlouhá 10-12 bytů, potvrzovací zprávy obsahují 2 byty navíc, kdyby nastala chyba a bylo třeba zaslat chybovou zprávu. Hlavička se skládá z těchto osmi částí:
    - Protocol ID – konstanta k identifikaci používaného protokolu, toto pole má délku 1 B.
    - Message Type – obecný typ zprávy s délkou 1 B, celkově mohou být čtyři typy zpráv.
        - 0x01 Job Request – je požadavek poslaný řídícím zařízením, může jít o čtení/zápis do paměti, čtení/zápis do bloků, spuštění/vypnutí zařízení a nastavení komunikace.
        - 0x02 Ack – jedná se o jednoduché potvrzení zprávy posílané slave zařízením bez pole s daty
        - 0x03 Ack-Data – potvrzení s daty, obsahuje odpověď na požadavek
        - 0x04 UserData – rozšíření původního protokolu, pole pro parametry, obsahuje ID požadavku/odpovědi (používají ji především programátoři pro debug, nastavení času nebo bezpečnostní funkce).
    - Reserved – je vždy nastaven na 0x0000 a jeho délka je 2 B.
    - PDU Reference – generováno masterem, zvyšováno s každým novým přenosem, používá se ke správnému přiřazení odpovědi k požadavku, pro toto pole se používá malá endianita, má délku 2 B.

    - Parameter Length – délka pole s parametry, využívá velkou endianitu a jeho délka je 2 B.
    - Data Length – délka pole s daty, využívá velkou endianitu a jeho délka je 2 B.
    - (Error Class) – přítomen pouze v Ack-Data zprávách pro případ chyby a má délku 1 B.
    - (Error Code) – přítomen pouze v Ack-Data zprávách pro případ chyby a má délku 1 B.
- Parameters – obsah parametrového pole se mění v závislosti na funkčním typu PDU a zprávě, kterou má přenášet.
- Data – volitelné pole pro data, pokud nějaká jsou (paměťové hodnoty, blok kódu, firmwarová data apod.) [31]

2.6 Modbus TCP/IP

Modbus TCP/IP byl prvním průmyslovým Ethernetem založen v roce 1999. Rozšiřuje základní Modbus protokol na Ethernet. Není považován za real-time protokol, a to i navzdory tomu, že původní Modbus je považován za velmi deterministický protokol, avšak nabízí real-time způsob komunikace. Umožňuje přenášení dat po různých sběrnicích a sítích (RS-232, RS-485, Ethernet TCP/IP). Používá princip klient-server komunikace, ta je podobná jako master-slave, avšak nabízí možnost ustavičného zasílání dat klientem serveru. Jedná se o otevřený a relativně jednoduchý protokol. Komunikace mezi Modbus TCP řídicím prvkem a Modbus TCP podřízeným prvkem funguje na čtyřech typech zpráv, jedná se o tyto zprávy: Modbus request, Modbus confirmation, Modbus indication, Modbus response.

- Modbus request je zpráva poslaná po síti klientem k inicializaci transakce
- Modbus indication je potvrzení o přijetí Modbus requestu ze strany serveru
- Modbus response je odpověď poslaná serverem
- Modbus confirmation je odpověď přijatá na straně klienta

Modbus TCP master zařízení má dvě možnosti komunikace, může komunikovat přímo s jedním, nebo všemi slave zařízeními v jeho síti, kdežto slave zařízení nemají možnost přímo komunikovat mezi sebou. Master zařízení (klient) posílá požadavky (request telegram, service request) na slave zařízení (server) a ten mu dle funkčního kódu zprávy zašle požadovaná data (response telegram), může nastat situace, že slave jednotka nemůže požadavek klienta zpracovat, v takovém případě odesílá místo odpovědi chybovou zprávu (error function code, exception response). Pro konfiguraci, monitorování a kontrolování vstupně/výstupních zařízení se využívají paměťové registry. Modbus TCP využívá Big-Endian formát, což znamená že nejdůležitější byte ze sekvence je uložen na nejnižší adrese (je první).

2.6.1 Modbusové funkce a registry

Modbusové registry

Tabulka 4: Seznam podporovaných registrů v Modbusu
Zdroj: vlastní zpracování

Odkaz	Typ	Název registrů	Popis
0xxxx (1-9999)	read/write	discrete output coils	0x reference adresy se používají pro vypsání digitálních výstupních kanálů
1xxxx (10001-19999)	read-only	discrete inputs contacts	On / Off status reference 1x je kontrolována korespondujícím digitálním vstupovým kanálem
3xxxx (30001-39999)	read-only	analog input registers	obsahují 16bitové číslo z externího zdroje (analogový signál)
4xxxx (40001-49999)	read/write	analog output holding registers	používá se k ukládání 16bitových numerických dat (binární nebo decimální), nebo k odesílání dat z CPU na výstupní kanály

Odkaz bereme jako jména lokací, tato čísla v samotné zprávě nenajdeme, jsou totiž nahrazena jejich adresami. Například první Holding registr, číslo 40001, má datovou adresu 0000. Rozdíl mezi těmito hodnotami je stanoven offsetem. Každá tabulka má jiný posun. 1, 10001, 30001, 40001. [32]

Modbusové funkce

Tabulka 5: Funkční zprávy podporované Modbusem
Zdroj: vlastní zpracování

Kód funkce	Funkce	Odkaz
01 (01H)	Read Coil (Output) Status	0xxxx
03 (03H)	Read Holding Registers	4xxxx
04 (04H)	Read Input Registers	3xxxx
05 (05H)	Force Single Coil (Output)	0xxxx
06 (06H)	Preset Single Register	4xxxx
15 (0FH)	Force Multiple Coils (Outputs)	0xxxx
16 (10H)	Preset Multiple Registers	4xxxx
17 (11H)	Report slave ID	

Zde vidíme všechny možné funkce, které v Modbus TCP master zařízení můžeme vložit do zprávy určené podřízeným jednotkám. Klient zažádá datové pole a poskytne serveru potřebná doplňující data potřebná ke správnému splnění akce dožadované klientským požadavkem. Datové pole typicky zahrnuje adresy registrů, čítací hodnoty a data. Pro některé typy zpráv nemusí toto pole existovat (má nulovou délku), jelikož ne všechny zprávy tato data vyžadují. Když zařízení fungující jako server (slave), odpovídá klientu (master), užívá k tomu „function code field“, což indikuje buď normální (error-free) odpověď, nebo hlásí, že se vyskytla příslušná chyba nebo výjimka (exception response). Normální odpověď jednoduše přeposílá originální funkční kód, zatímco chybná odpověď pošle funkční kód, kde bude kód změněn přepsáním nejdůležitějšího bitu (most significant bit) na logickou 1. Například Read Holding Registers funkce má funkční kód (0000 0011, 2), (03, 8), bez chyby se vrátí tento kód beze změny, ovšem

pokud nastane chyba, tato zpráva se změní na (1000 0011, 2), (13, 8), takto je pak poslán v „function code field“, kam je také přidán unikátní kód chyby, který klientovi (masterovi) definuje, co za chybu nastalo, nebo důvod proč nastala.

Klientský aplikační program musí z tohoto důvodu umět zvládat výjimky (exception handling), jedním způsobem je se pokoušet zasílat originální zprávu a čekat na správnou odpověď, zasláním diagnostického dotazu nebo jednoduché oznámení výskytu chyby obsluze. [33]

Modbusové výjimky

Tabulka 6: Seznam výjimkových zpráv, jež mohou v Modbus komunikaci nastat
Zdroj: vlastní zpracování

Kód	Výjimka	Popis
01	Illegal function	Funkční kód obdržený v dotazu není povolen nebo není validní.
02	Illegal data adress	Datová adresa obdržená v dotazu není slavem (serverem) povolená hodnota nebo není validní.
03	Illegal data value	Hodnota obsažená v dotazovém datovém poli není povolená slavem nebo není validní.
04	Slave/Server device failure	Slave (server) selhal během exekuce programu. Neobnovitelná chyba nastala, když se slave (server) snažil provést požadovanou funkci.
05	Acknowledge	Slave (server) přijal požadavek a zpracovává, k jeho zpracování ovšem potřebuje moc času. Tato výjimka je poslána, aby se předešlo timeout chybám na straně mastera (klienta).
06	Slave/Server device busy	Slave (server) zpracovává časově náročnou akci. Master (klient) by měl vyčkat a přeposlat požadavek, když je slave zařízení volné.

07	Negative acknowledge	Slave (server) nemůže vykonat funkci žádanou v dotazu. Master (klient) by měl vyžádat diagnostickou informaci ze zařízení, které vrátilo chybu.
08	Memory parity error	Slave (server) se pokusil číst prodlouženou paměť, ale narazil na paritní chybu v paměti. Master (klient) může zkusit znovu poslat požadavek, ale může být vyžadován servis na straně slave.
0A	Gateway problem	Výchozí brána není přístupná
0B	Gateway problem	Cílené zařízení neodpovědělo, tato výjimka je generovaná výchozí bránou.
FF	Extended exception response	Výjimkové PDU obsahuje více informací o výjimce, posílá se navíc pole o délce 2 bytů, které zasílá bytovou délku, která je třeba k popisu nastané chyby.

2.7 Ethernet PowerLink

Vyvíjen organizací Ethernet PowerLink Standardization Group je open-source protokol, vyvinut za účelem vytvoření velmi rychlé sběrnice na platformě FastEthernet s deterministickými odezvami a minimálním rozptylem časování telegramů, kromě cyklické výměny dat podporuje komunikaci acyklickou, která nijak neomezí komunikaci cyklickou. Staví na dvou nejnižších vrstvách (fyzická, linková) referenčního modelu ISO/OSI. Je navržen podle standardu FastEthernet odpovídá mu topologií, provedením fyzické vrstvy a rychlostí (100 Mb/s). K adresaci využívá MAC adresy. Prvky jeho sítě se skládají z Managing node (MN) a Controlled nodes (CN), řídicí uzel, řízené uzly a z opakovacích jednotek, což jsou jen posilovače signálu nebo rozdělovače.

- MN – implementuje automatizační a kontrolní úlohy (řídicí prvek například PLC), stará se o síťovou komunikaci a v komunikační sítí Ethernet PowerLink se vyskytuje pouze jeden.
- CN – jsou ostatní zařízení s komunikačními vlastnostmi, spadají mezi ně například senzory, celkový počet těchto zařízení je omezen na 240 řízených uzlů na jeden řídicí uzel.
- Switch/Hub – aktivní prvek sítě pro větvění, nebo posilování signálů.

Základní cyklus Ethernet PowerLinku probíhá takto, po start-up sekvenci začne real-time doména pracovat dle předem stanovených podmínek, rozvrh základního cyklu je kontrolován Managing nodem (master zařízením), celková časová náročnost cyklu závisí na množství Izochronních dat, asynchronních dat a počet adresovaných zařízení v cyklu. Základní cyklus sestává z těchto fází:

- Start phase – řídící zařízení vyšle synchronizační zprávu přes všechna zařízení, tento rámec se nazývá Start of Cycle (SoC).
- Isochronous phase – řídící zařízení zavolá každé zařízení a dotáže se, aby mu zaslalo časově kritická data pro proces nebo ovládání pohybu posláním Poll Request (PReq) rámce, tato adresovaná zařízení odpoví pomocí Poll Response (PRes) rámce. Během této fáze naslouchají všechna zařízení všem datům, která se na síti v této fázi vyskytují.
- Asynchronous phase – řídící zařízení dá práva právě jednomu zařízení, pro zasílání ad-hoc dat, což mezi těmito zařízeními vytvoří komunikační kanál, zasláním Start of Asynchronous rámce a adresované zařízení odpoví. IP standardově orientované protokoly a adresování je v této fázi povoleno.

Kvalita real-time komunikace závisí na přesnosti celkového základního času cyklu. Délka jednotlivých fází se může lišit, dokud celkový fázový čas spadá do časových hranic pro celý cyklus, tento cyklový čas je monitorován řídícím zařízením, časy jednotlivých fází se dají konfigurovat. [34]

2.8 EtherNet/IP

Vyvíjen společnostmi Allan Bradley (Rockwell Automation) a Open DeviceNet Vendors Association je otevřený průmyslový standard, který byl prvně prezentován v roce 2001. Byl navržen tak, aby plně spolupracoval s klasickým Ethernetem a využívá pro něj definovaný komunikační model (beze změn na první až čtvrté vrstvě referenčního modelu ISO/OSI) a topologie. Je jednou ze čtyř sítí, která rozšiřuje CIP na průmyslovou síť. Jeho komunikace není primárně zaměřena cyklicky ale časově, je proto nutné, aby kontrolní příkazy byly včas doručeny. Každý uzel v síti má předem definovaný typ se specifickým účelem, tato definice zařízení se nazývá profil. Může koexistovat s klasickou ethernetovou komunikací na jedné síti. Standard EtherNet/IP definuje tři typy zařízení:

- Messaging class – takto definované zařízení podporuje explicitní komunikaci bez možnosti implicitní, typicky se jedná o zařízení ke konfiguraci nebo diagnostiku.
- Adapter class – takto definované zařízení zpracovává data v real-time režimu, nemůže však samo navazovat spojení, typicky se jedná o vstupně/výstupní zařízení.
- Scanner class – takto definované zařízení navazuje spojení pro datový přenos v reálném čase se zařízeními Adapter class nebo s ostatními zařízeními jejich typu, typicky se jedná o řídicí prvky v síti například PLC. [35]

K výměně dat využívá EtherNet/IP oba protokoly transportní vrstvy TCP/IP a UDP/IP vzhledem k tomu, že CIP využívá customer-producer, je tato architektura použita i pro komunikaci v EtherNet/IP. Protokol CIP rozlišuje komunikaci na cyklickou implicitní, která slouží k přenosu vstupně/výstupních zpráv, ty jsou zprostředkovány užitím UDP/IP protokolů a explicitní dotaz/odpověď telegramy ke konfiguraci a získávání dat mezi dvěma uzly sítě, tato komunikace probíhá za použití UDP/IP protokolů. [36]

Pro zahájení spojení je ze žadatele (originator) vyslaná explicitní zpráva o požadavku k vytvoření spojení (Forward_Open) cílovému uzlu (target), v této zprávě jsou

obsaženy návrhy parametrů. Pokud je cílové zařízení schopné spojení navázat, je zasláno potvrzení, které obsahuje již konečné parametry, za kterých je spojení navázáno. Mezi spojovací parametry patří: Identifikátor, způsob přenosu, spouštěcí mechanismus komunikace, počet a formát dat. Identifikátor (Connection ID) slouží k jasné identifikaci spojení, a je různý pro oba směry přenosu, způsob přenosu vyplívá z typu a způsobu odeslání dat. Spouštěcím mechanizmem rozumíme například změnu stavu nebo cyklickou komunikaci. Pokud je počet přenášených dat nulový, je funkce zařízení správná a proběhne výměna dat.

EtherNet/IP není přímo určen pro úlohy v reálném čase, pro zprostředkování real-time vlastností používá pouze základní mechanismy a spoléhá se na rychlost Ethernetu. [35] [37]

2.9 Další způsoby komunikací (OPC, I/O Link)

2.9.1 OPC

OPC je služba, jejíž cílem je vytvořit jednotné komunikační rozhraní mezi hardwarem různých výrobců a softwarovými produkty průmyslové automatizace, používá klient-server architekturu, kde klient i server jsou čistě softwarové aplikace.

- Klient – přijímá data z OPC serveru v definovaném protokolovém formátu a prezentuje tato data uživateli (HMI, grafy, reporty).
- Server – komunikuje s připojenými zařízeními jejich komunikačním protokolem (např. Modbus), získaná data převádí do OPC formátu a poskytuje je nadřazeným aplikacím.

Díky OPC je možné do projektu zařadit zařízení a aplikace různých výrobců nehledě na komunikační rozhraní. OPC protokol je definován organizaci OPC Foundation prostřednictvím takzvaných OPC specifikací, tato definice je volně přístupná technická dokumentace.

Bez OPC je nutné pro každé zařízení mít speciální ovladač pro čtení/zápis dat do tohoto zařízení a v případě více ovladačů může docházet k vzájemnému ovlivňování

komunikace nebo k nekompatibilitě s operačním systémem. Při potřebě přidání dalšího zařízení je třeba úprava řídícího systému.

S OPC tyto starosti odpadají, jelikož jediné komunikační rozhraní mezi hardwarovými a softwarovými systémy je OPC a společný komunikační kanál je zpravidla podniková síť (Ethernet). Dochází i k zjednodušení přidávání dalších stanic.

2.9.2 Příklady architektury OPC klient-server

Jednoduchá aplikace na lokální PC stanici

Oba programy (server + klient) jsou naistalovány na jedné stanici. Tento způsob je typický pro jednoúčelové aplikace například monitorování čerpací stanice, výrobního stroje, v případě potřeby je možné počítač připojit k síti a rozšířit o další OPC prvky

Jednoduchá aplikace v rámci Ethernetové sítě

OPC klient a server jsou zvlášť nainstalované na jiných stanicích, jednoduché přidání dalších OPC klientů/serverů.

Rozsáhlá aplikace OPC

V projektu jsou čtená data z více OPC serverů a jsou zpracovávána ve více klientských PC stanicích, zpravidla tento systém najdeme ve velkých podnicích, kde operátoři monitorují na svých počítačích celé výrobní linky, management sleduje stav výroby, plánovači sledují plnění plánů výroby, pracovníci kvality plnění kvality výroby. Běžně se setkáváme s projekty do deseti OPC serverů, zhruba stejného počtu klientů a s několika stovkami (až tisíci) přenášenými veličinami. [38]

2.9.3IO-Link

Další možností realizace průmyslové komunikace je IO-Link. Ten však není klasickou průmyslovou sběrnicí, jedná se o způsob propojení vstupně-výstupních modulů pomocí jednoho tří nebo čtyř žílového média do takového zařízení, které slouží jako mezivrstva pro master-slave komunikaci, v této mezivrstvě dochází k transformaci signálu do takového tvaru, který je vyžadován pro komunikaci s řídícím zařízením, zařízením této mezivrstvy se nazývá IO-Link Master. Příkladem takových zařízení jsou Ether-

netové IO moduly, ty komunikují se vstupně-výstupními zařízeními pomocí IO-Linkového připojení a tuto komunikaci převádí na ethernetovou, skrz ethernet je poté realizována komunikace s řídícím zařízením.

Protože se jedná o snadné řešení, získává oblíbenost jak u výrobců, tak u servisních techniků, proto modulů pro IO-Link existuje několik a podporuje značnou část průmyslových sběrnic. Jednou z jeho největších výhod je jednoduchost, jeho zavedení do průmyslové sítě je časově nenáročné a díky jednotnému přenosovému také omezuje obvykle časově náročné zavádění kabelů.

IO-Link podporuje tři typy zprávy. Process Data, která mohou obsahovat nejméně 1 bit a nejvíce až 32 bytů dat s časem cyklu kolem 2ms. Service Data umožňují získání detailních informací o zařízení, které obdrželo tento typ zprávy jako konfiguraci, jméno zařízení, typ zařízení, sériová čísla, status, detailní diagnostiku a podobně. Events jsou posledním typem zpráv, označuje takové události, které je potřeba nahlásit co nejrychleji. Tyto události se běžně nestávají, a proto nejsou zahrnuta v Process Data, označují například alarmy.

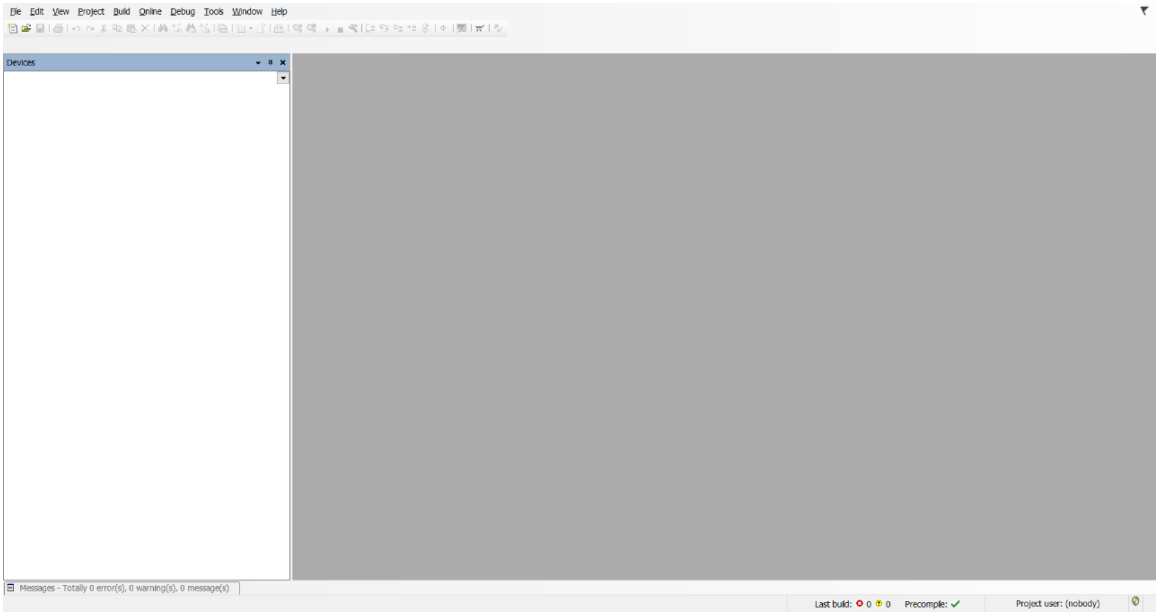
Každé IO-Linkové zařízení obsahuje takzvaný IO-Link Device Description File, který obsahuje informace, které toto zařízení popisují jako jeho sériové číslo, jméno a identifikátor výrobce, podporované přenosové rychlosti a popis dat, která zařízení poskytuje. [39] [40] [41]

3. Použité technologie

3.1 CODESYS

CODESYS je softwarová platforma vyvíjená v Německu pro tvorbu aplikací řešících problémy průmyslové automatizace založená na CoDeSys programovacím standardu IEC 61131-3. Vývojářům poskytuje rozsáhlá integrovaná řešení pro pohodlné projektování automatizačních aplikací. Jedná se o globálně rozšířené vývojové prostředí s velkou podporou výrobců automatizačních zařízení. Jeho služeb využívá i několik univerzit, kde pomocí něj vyučuje programování automatizačních aplikací. Podporuje šest programovacích jazyků Instruction List, Structured Text, Ladder Diagram, Squential Function Chart, Function Block Diagram a Continuous Function Chart. Pro testovací účely byl využit CODESYS V3.5 Service Pack13.

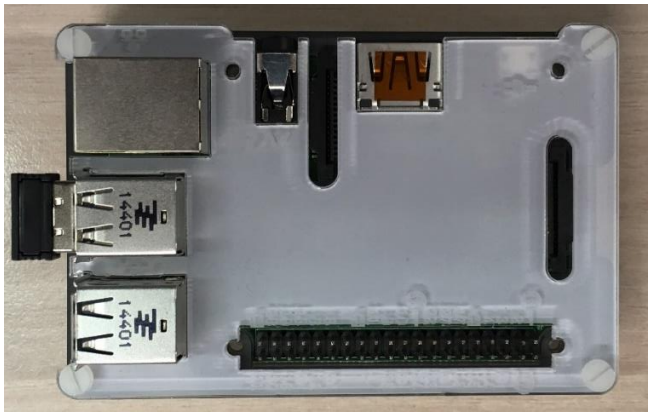
CODESYS má spoustu důležitých funkcí, ale nejdůležitější pro účel simulací je možnost spuštění virtuálního PLC v počítači, díky tomuto je možné s počítačem nakládat jako s master nebo slave zařízením, podle potřeby aplikace. [42] [43]



Obrázek 2: Základní prostředí CODESYS
Zdroj: vlastní zpracování

3.2 Raspberry Pi

Raspberry Pi je levný, malý počítač určený především jako vývojový a edukační kit. Není to zařízení vhodné do průmyslu. K funkčnosti potřebuje operační systém nainstalovaný na SD kartě, jako OS především používá distribuci Linuxu, která je vyvinuta přímo pro Raspberry Pi - Raspbian. Je velice oblíbený kvůli jeho široké škále využití a nízké ceně ($35). Pro naše účely bylo využito Raspberry Pi 2 Model B 1GB s instalovaným Raspbianem. Do Raspberry byl nainstalován CODESYS runtime, pomocí poskytovaného CODESYS Control for Raspberry Pi SL balíčku, který z něj učinil fungující PLC zařízení. Bylo nutné povolit SSH a nastavit statickou IP adresu odpovídající masce užité sítě.

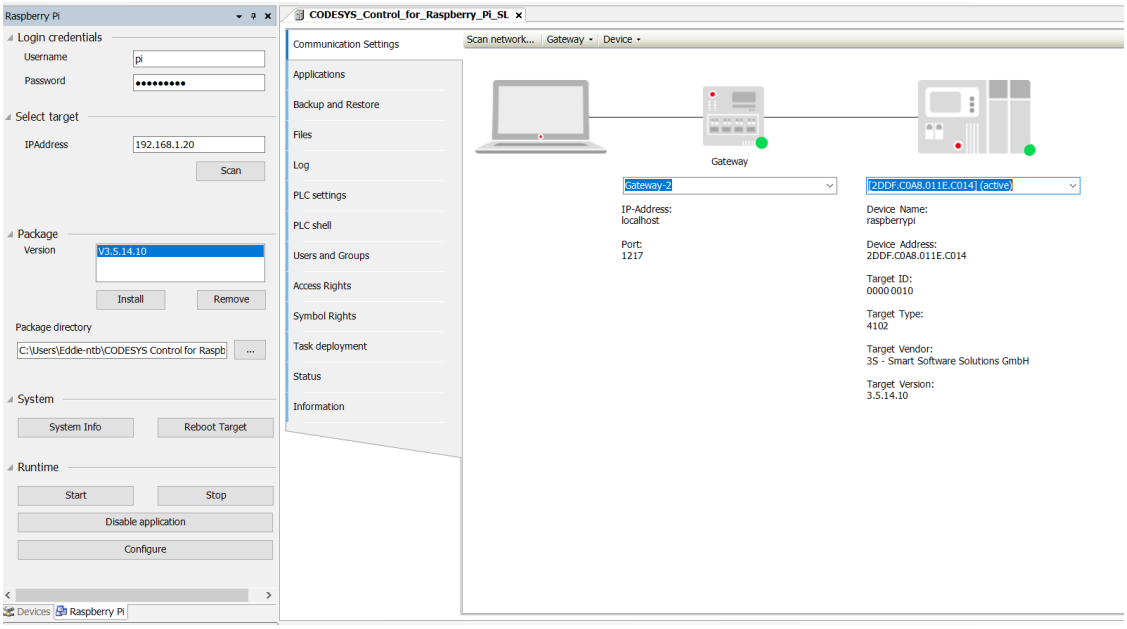


Obrázek 3: Raspberry Pi 2 Model B 1GB
Zdroj: vlastní zpracování

Ke kontrole funkčnosti Raspberry je zapotřebí monitoru, nebo programu PuTTy, který se na něj připojí pomocí SSH. [44]

3.2.1 Připojení Raspberry Pi s CODESYS

Pro propojení bylo nutné síťového spojení Raspberry a počítače s funkčním CODESYS prostředím, do kterého se nainstalovala poskytnutá knihovna pro řízení Raspberry. CODESYS se poté pomocí SSH připojil na zařízení a nainstaloval runtime.



Obrázek 4: Instalace CODESYS runtime do Raspberry Pi
Zdroj: vlastní zpracování

3.3 WireShark

WireShark je počítačová aplikace k analyzování síťových paketů, zachytává síťovou komunikaci a poskytuje náhled na její strukturu a detailní náhled na pakety. Představme si WireShark jako měřící zařízení, které kontroluje, co se na síti děje. WireShark je open-source nástroj, který má velkou komunitní podporu a jedná se o jeden z nejlepších síťových analyzačních nástrojů dnes.

V této práci byl použit na pozorování struktury komunikace jednotlivých průmyslových sběrnic a na poskytnutí náhledu na strukturu jednotlivých komunikačních protokolů užívaných sběrnicemi. [45]

3.3.1 WinPcap

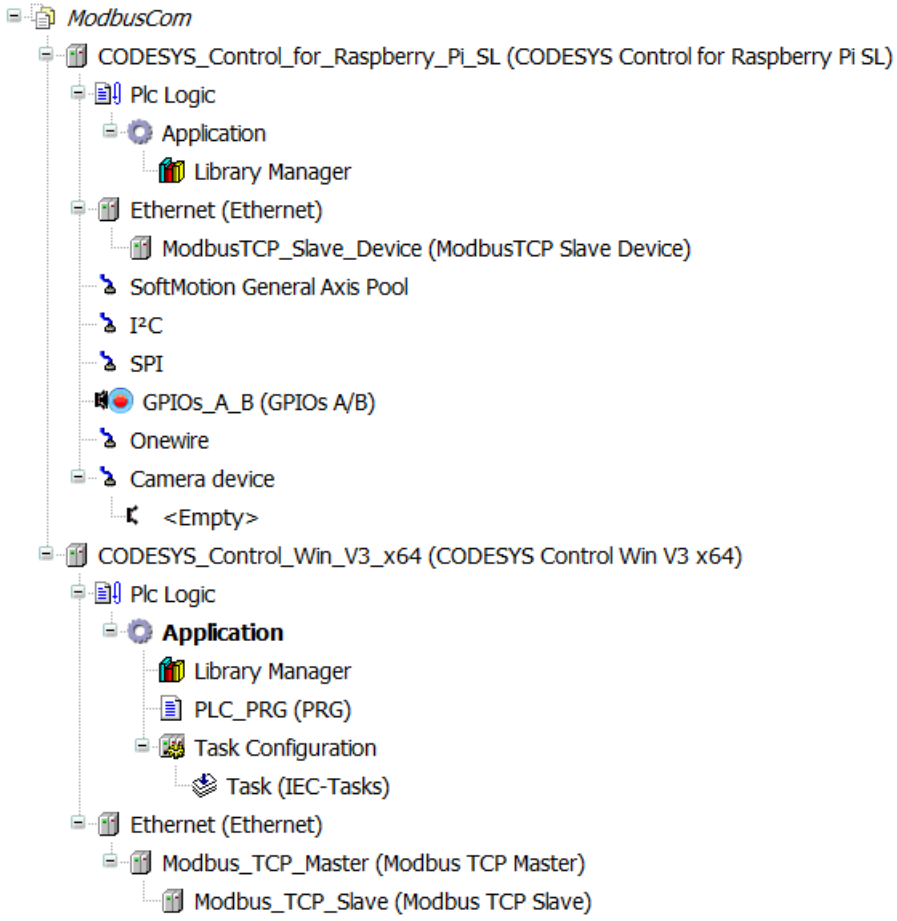
WinPcap je nástroj pro Windows, který umožňuje zachytávání komunikace na úrovni linkové vrstvy, umožňuje aplikacím zachytávat a přenášet síťové pakety mimo základní protokolový zásobník. [46]

4. Simulace

4.1 Modbus TCP

Pro zprovoznění simulace mezi dvěma zařízeními za použití Modbus TCP protokolu nebylo potřeba žádného dalšího hardwaru, pouze propojené Raspberry Pi a počítače s CODESYS.

4.1.1 Struktura projektu

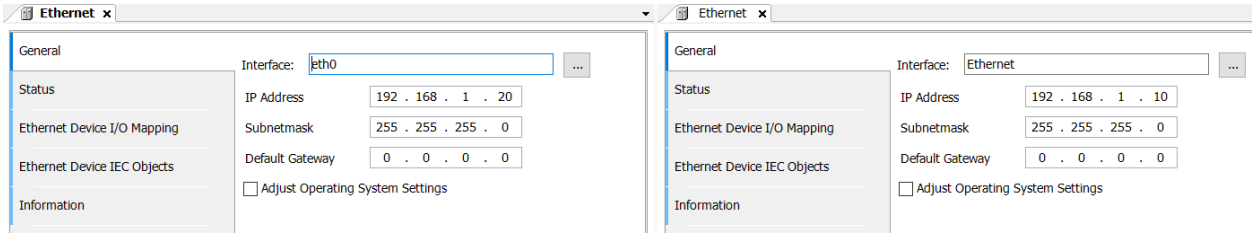


Obrázek 5: Struktura projektu komunikace Modbus TCP
Zdroj: vlastní zpracování

Na schématu projektu je vidět, že pro obě zařízení jsou definována síťová rozhraní, ta obsahují informace o rozhraní, které je používáno pro realizaci spojení. Také je vidět, že Raspberry Pi v tomto projektu slouží jako Slave zařízení, a virtuální PLC na počítači jako Master.

4.1.2 Nastavení Modbus TCP simulace

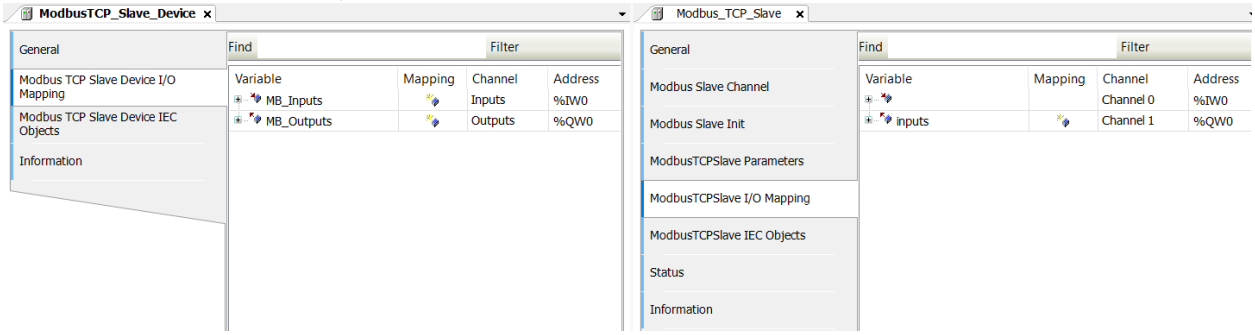
Nastavení ethernetového rozhraní



Obrázek 6: Nastavení ethernetových rozhraní
Zdroj: vlastní zpracování

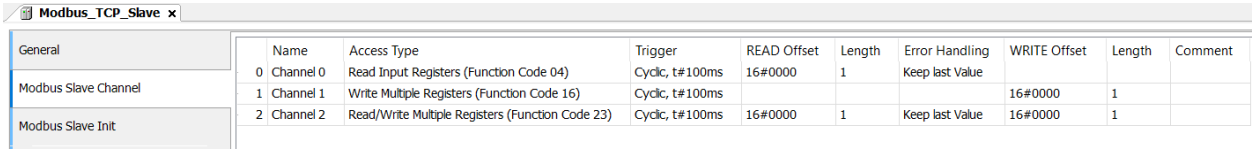
Zde je vidět nastavení jednotlivých rozhraní, rozhraní musí být aktivní a funkční, aby ho CODESYS mohl najít a pracovat s ním.

Mapování a kanály



Obrázek 7: Nastavení mapování Modbus TCP komunikace
Zdroj: vlastní zpracování

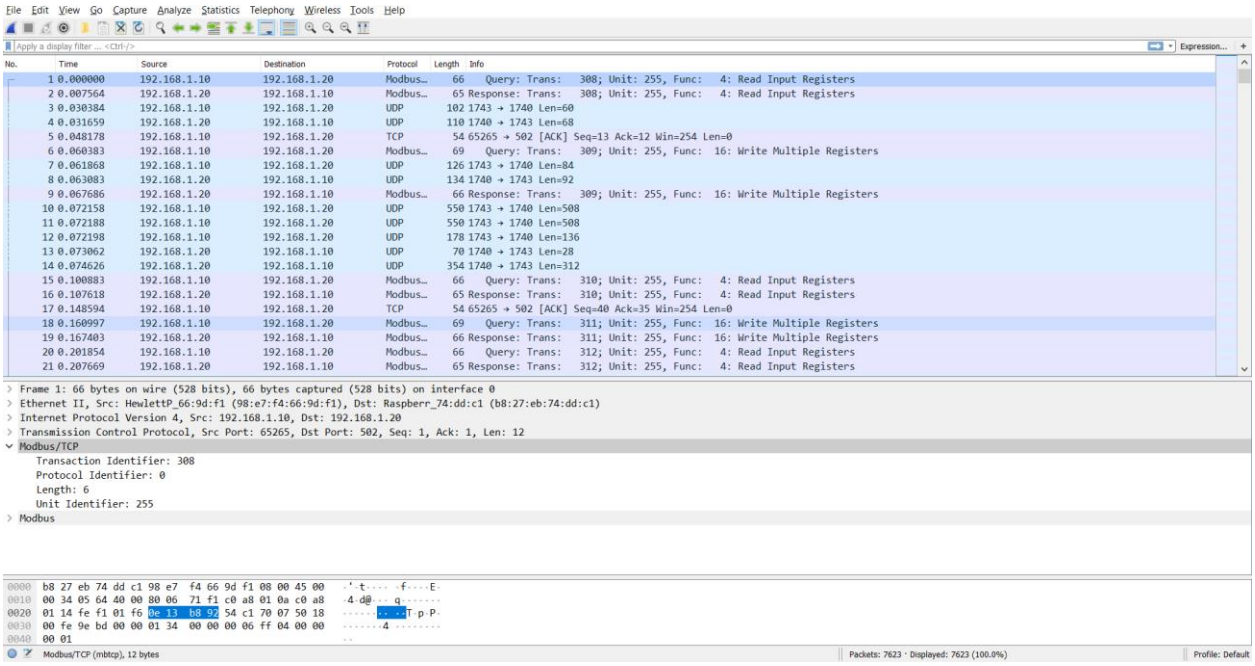
Na obrázku 7 je vidět nastavení pro mapování jednotlivých komunikačních kanálů, toto nastavení definuje jména proměnných, ke kterým pak lze v programu přistupovat.



Obrázek 8: Nastavení komunikačních kanálů
Zdroj: vlastní zpracování

Nastavení komunikačních kanálů určuje, jakým způsobem může zařízení přijímat nebo odesílat data, nastavují se zde všechny důležité parametry komunikace.

4.1.3 Komunikace ve WireShark



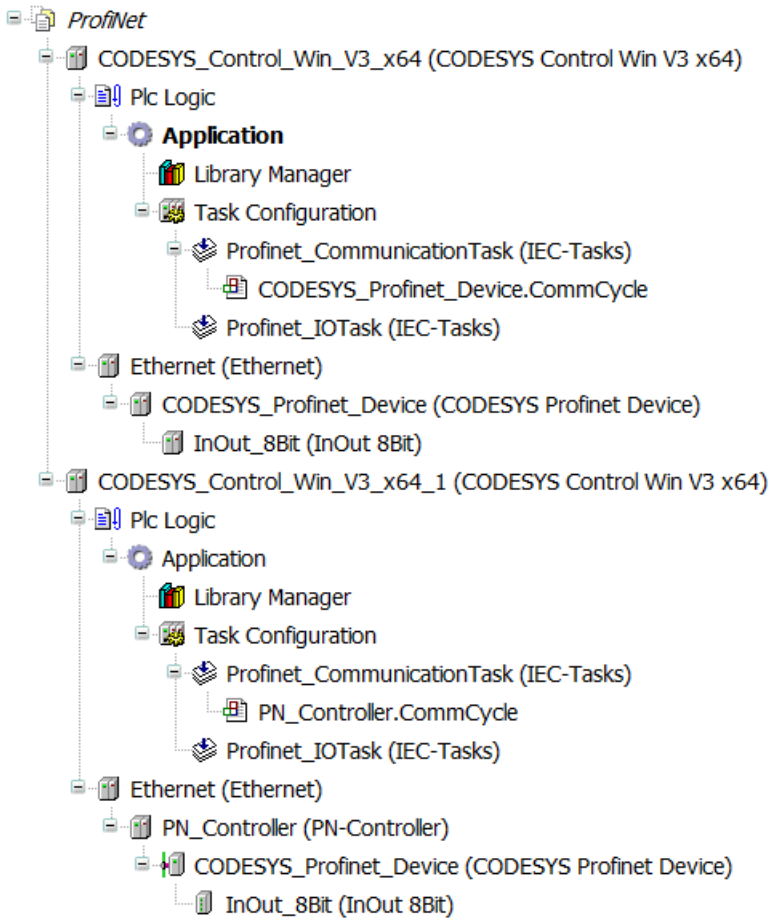
Obrázek 9: Modbus TCP komunikace ve WireSharku
Zdroj: vlastní zpracování

V průběhu jednoduché datové transakce byla komunikace z aktivního rozhraní zachytávána programem WireShark. Můžeme v ní vidět zahajovací i potvrzovací zprávy od vysílacího a přijímacího zařízení. Tyto zprávy mají vždy stejnou funkci a funkční číslo.

4.2 ProfiNet

Pro zprovoznění simulace mezi dvěma zařízeními za použití ProfiNet protokolu nebylo potřeba žádného dalšího hardwaru, avšak nepovedlo se mi tuto simulaci uskutečnit mezi počítačem a Raspberry Pi, bylo tedy potřeba použít dvou počítačů s instalovaným CODESYS a jejich dvou virtuálních PLC. Pro ProfiNet simulaci bylo potřeba změnit konfiguraci virtuálního PLC a přidat systémový ethernet, další nutností bylo nainstalovat WinPcap.

4.2.1 Struktura projektu



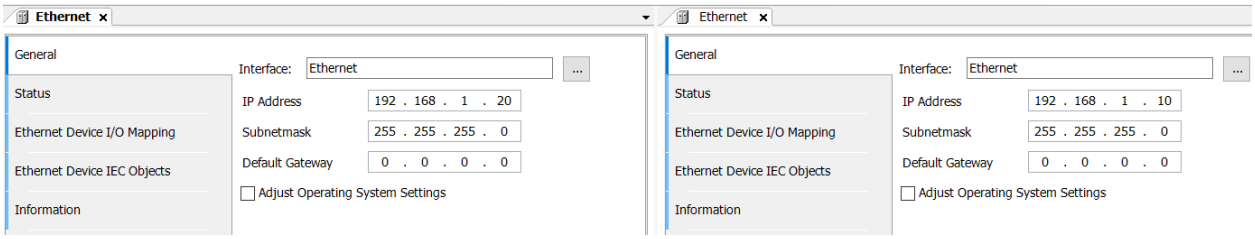
Obrázek 10: Struktura projektu komunikace ProfiNet
Zdroj: vlastní zpracování

Ze schématu je patrné že jedno zařízení funguje jako ProfiNet Controller (master) a druhé jako ProfiNet Device (slave). Každé zařízení má nadefinované síťové rozhraní, které daná zařízení propojuje.

4.2.2 Nastavení ProfiNet simulace

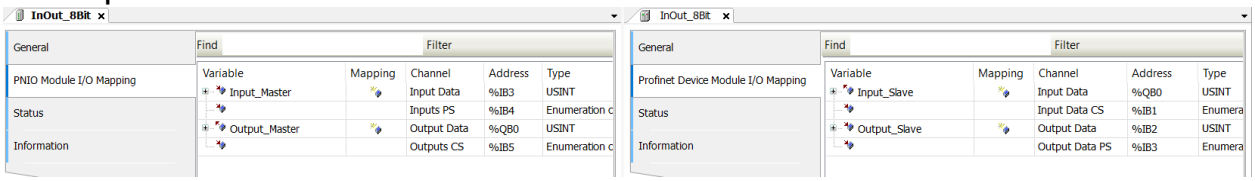
Nastavení ethernetového rozhraní

Nastavení ethernetového rozhraní je stejné jako pro Modbus TCP s rozdílem, že se nejedná o síťové rozhraní Raspberry Pi, ale klasického počítače.



Obrázek 11: Nastavení ethernetových rozhraní
Zdroj: vlastní zpracování

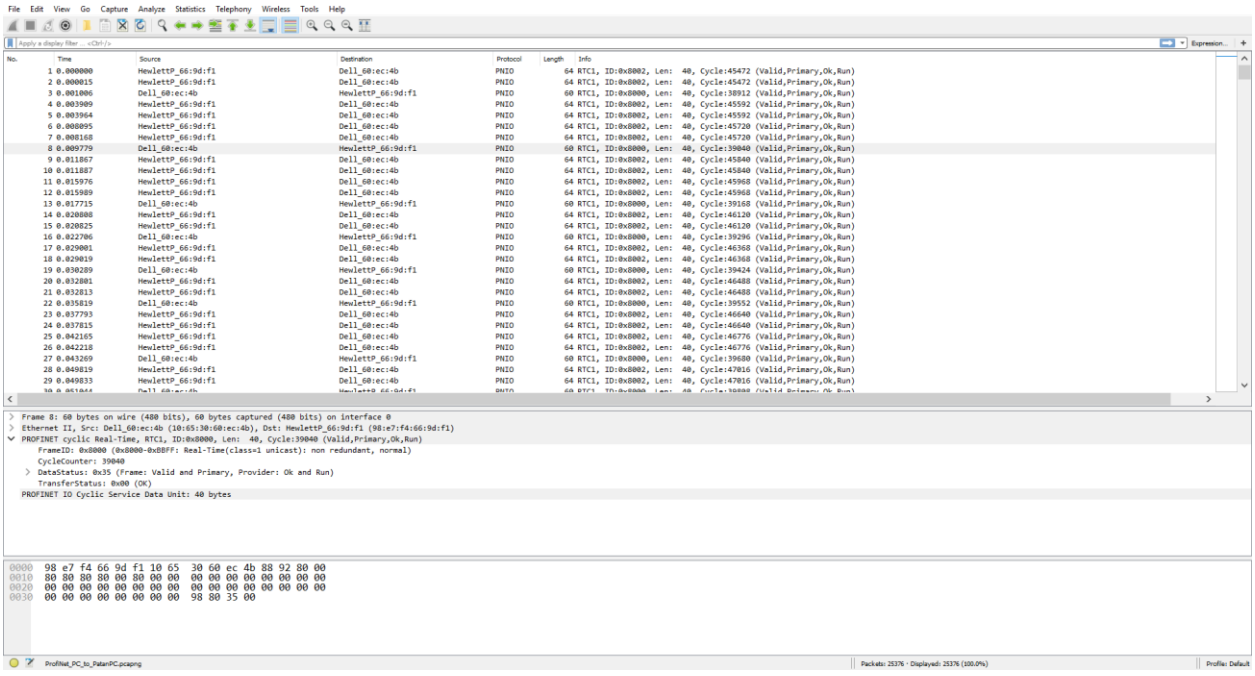
Mapování



Obrázek 12: Nastavení mapování ProfiNet komunikace
Zdroj: vlastní zpracování

Na tomto obrázku je vidět nastavení pro mapování jednotlivých komunikačních kanálů, toto nastavení definuje jména proměnných, ke kterým pak lze v programu přistupovat.

4.2.3 Komunikace ve WireShark



Obrázek 13: ProfiNet komunikace ve WireSharku
Zdroj: vlastní zpracování

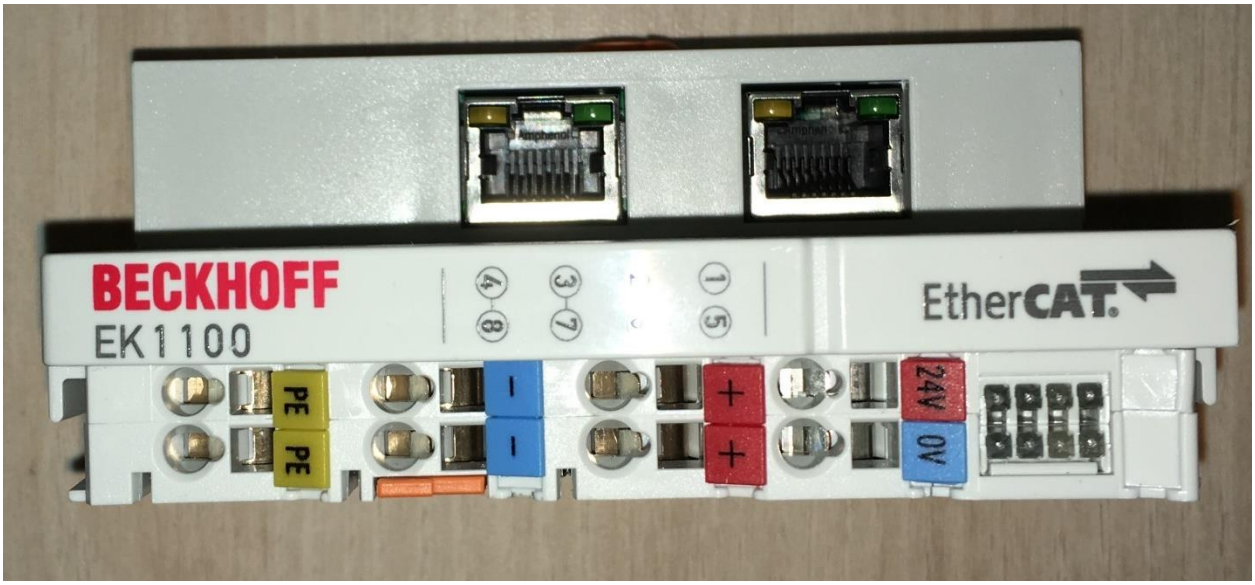
V průběhu jednoduché datové transakce byla komunikace z aktivního rozhraní zachytávána programem WireShark. Tato znázorněná zpráva pochází z master zařízení, jedná se o cyklickou, real-time zprávu.

4.3 EtherCAT

K zprovoznění komunikace za použití EtherCAT sběrnice už nestačí mít jen dvě virtuální PLC, je potřeba EtherCAT master zařízení, které bude řídit komunikaci, kvůli tomuto bylo napůjčováno několik zařízení.

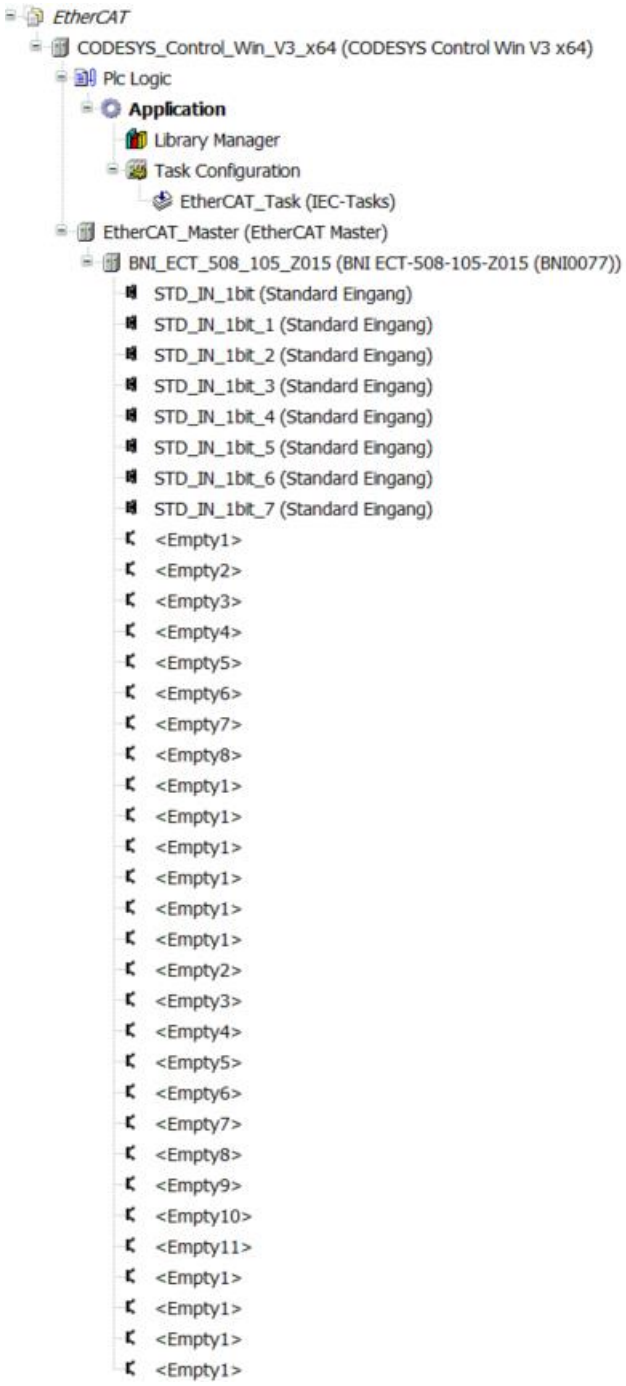
4.3.1 Zařízení

Obrázek 14: EtherCAT IO-Link Master zařízení od firmy Balluff, s IO-Linkovými vstupy
Zdroj: vlastní zpracování



Obrázek 15: EtherCAT Coupler of firmy Beckhoff
Zdroj: vlastní zpracování

4.3.2 Struktura projektu



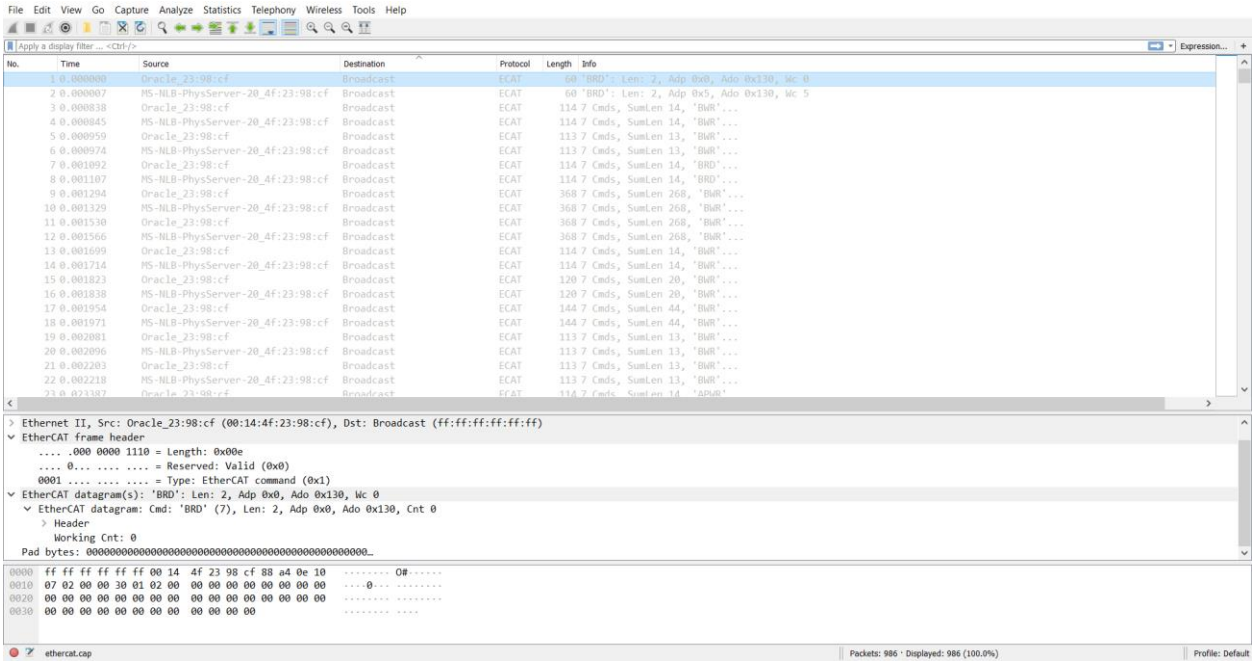
Obrázek 16: Struktura projektu EtherCAT komunikace
Zdroj: vlastní zpracování

Ze schématu struktury projektu je patrné, že je zde pouze použito jedno virtuální PLC odkazující na EtherCAT IO-Link master zařízení. Slave zařízení není definováno

z toho důvodu, že EtherCAT master se stará o IO-Linkové zařízení k němu připojené, a tudíž není třeba je definovat.

Bohužel se mi tuto komunikaci nepodařilo funkčně nasimulovat, proto příklad komunikace z WireShark není z mé simulace. Příklad simulace je z WireSharkové dokumentace. [47]

4.3.3 Komunikace ve WireShark



Obrázek 17: EtherCAT komunikace ve WireSharku
Zdroj: vlastní zpracování

5. Porovnání průmyslových sběrnic

Zásadní otázkou při volbě sběrnic je: Dle jakých faktorů vybrat vhodnou průmyslovou sběrnici?

Když nebude uvažován fakt takových parametrů, jako je například historie používání jedné sběrnice, závislost na dodavateli nebo zákaznické preference, tak zůstává několik vlastností, na jejichž základě by měl být sběrnicový systém vybrán.

Jako nejdůležitější lze považovat typ aplikace, protože na něm záleží požadavky na sběrnici kladené.

Modbus TCP se nehodí na striktně real-time úlohy, zatímco EtherCAT je přímo vyvíjen na tyto typy úloh. ProfiNet lze zařadit na pomezí mezi tyto dvě sběrnice co se možnosti real-time komunikace týká.

Dalším důležitým aspektem je zajisté podpora použitých zařízení, tento parametr je ve výsledku také o ceně. Zařízení existuje velké množství a pro každou sběrnici existuje možnost, jak zařízení nakonfigurovat či upravit tak, aby plnilo požadovanou funkci. Tento parametr se poté prolíná s požadavkem velikosti úlohy.

Velikostí úlohy se rozumí počet řídicích a řízených zařízení, pokud se jedná o úlohu s menším počtem zařízení, kde není potřeba řízení v reálném čase, Modbus TCP je vhodná volba. V případě potřeby větší rychlosti přenosu dat, tak je vhodnější spíše ProfiNet nebo EtherCAT. Když se bude jednat o větší aplikace, je EtherCAT nejvhodnější volbou.

Jako další aspekt lze také uvést náročnost na montáž. V tomto může být většina, především starších sběrnic časově náročná – např. nutnost instalace CAN konektorů. Vzhledem k možnosti EtherCATu napájet a komunikovat jedním kabelem (EtherCatP), je v tomto ohledu favoritem. Nutno podotknout, že hledisko množství kabeláže a náročnost instalace není často zohledňována, ale např. u robotických aplikací to způsobuje problémy.

Skutečnost existence sběrnice IO-Link dokazuje, že tento aspekt je velmi důležitý a komplexní, kombinace průmyslových sběrnic s IO-Linkem poskytuje možnost, jak tento problém co nejvíce zjednodušit.

6.Způsoby propojení desktopových aplikací

Propojení desktopové aplikace s průmyslovými zařízeními se provádí pomocí knihoven, které jsou naprogramované, aby podporoval komunikaci za použití průmyslových protokolů, existuje jich několik v různých programovacích jazycích.

6.1 Python-snap7

Tento balíček pro Python rozšiřuje funkcionalitu existující knihovny Snap7 pro python. Slouží převážně pro proprietární S7 komunikaci mezi PLC zařízeními od firmy Siemens, dá se ovšem použít i pro komunikaci přes průmyslovou sběrnici ProfiNet.

Pro použití tohoto balíčku je nutné mít nainstalovanou Snap7 knihovnu, která je ke stažení z oficiálních stránek knihovny a poté samotný balíček, ten se instaluje příkazem `pip install python-snap7` v příkazové řádce v adresáři Pythonu. [48]

6.2 EasyModbusTCP/UDP/RTU

EasyModbusTCP/UDP/RTU je knihovna, která poskytuje funkcionalitu datového přenosu pomocí protokolu Modbus, je dostupná ve třech verzích, jedna pro .NET aplikace, druhá pro JAVA a třetí pro Python. Podporuje všechny Modbusové komunikační způsoby. [49]

K instalaci této knihovny pro Python je potřeba pouze instalovat balíček, který tuto knihovnu rozšiřuje pro Python, a to příkazem `pip install easymodbus` v příkazové řádce v adresáři Pythonu.

6.3 Simple Open EtherCAT Master Library

SOEM je EtherCAT master knihovna vyvíjená v jazyce C, jejíž účelem je poskytovat funkce datového přenosu za použití průmyslové sběrnice EtherCAT. Je volně dostupná k užití a distribuci pod licencí GNU. [50]

7. Závěr

Tato Bakalářská práce měla několik hlavních cílů, z nichž prvním bylo získání teoretických znalostí a zjištění možností průmyslových sběrnic založených na ethernetu. Díky velkému množství dokumentů a dobře popsaným dokumentacím bylo možné pochopit principy funkce jednotlivých sběrnic a na základě toho poznat jejich výhody a nevýhody.

V průběhu práce jsem se důvěrně seznámil s vývojovým prostředím CoDeSys, které je velmi rozšířené a nabízí širokou paletu možností a podpory napříč různými výrobci průmyslových zařízení, mezi tato zařízení spadá i využité Raspberry Pi, které bylo také velice užitečným přínosem a získanou zkušeností.

Díky získaným znalostem bylo možné vytvoření projektů sloužících jako simulace komunikace dvou ze tří zvolených průmyslových sběrnic. Díky vytvořeným simulacím mi byl poskytnut náhled na živou výměnu dat mezi dvěma průmyslovými zařízeními, což mi poskytlo ještě lepší povědomí o způsobu funkce protokolů, jež vybrané sběrnice využívají ke komunikaci.

V průběhu vybírání existujících knihoven určených k průmyslové komunikaci jsem zjistil, že podpora a vývoj nástrojů řešících tento specifický problém je velmi rozšířená a pro většinu průmyslových sběrnic už existuje přinejmenším jeden způsob k realizaci tohoto typu úlohy.

Nabyté teoretické i praktické znalosti budou využity pro tvorbu aplikace, která bude nasazena v prostředí průmyslové automatizace. Tato aplikace bude umět komunikovat po zvolené průmyslové sběrnici bez nutnosti doplňujících zařízeních a předávat data z výroby do klasických počítačů, díky tomuto se vyřadí jedna vrstva potřebná k diagnostice a kontrole stavu zařízeních.

Citovaná literatura

[1] „Industrial communications,“ [Online]. Available: https://idboxrt.com/en/industrial-communications/. [Přístup získán duben 2019].

[2] „Interfaces and bus systems: The right communication for the,“ [Online]. Available: https://www.hbm.com/en/3237/interfaces-and-bus-systems-the-right-communication-for-the-industrial-sector/. [Přístup získán duben 2019].

[3] „ISO 1189:1986,“ září 1986. [Online]. Available: https://www.iso.org/standard/5782.html. [Přístup získán duben 2019].

[4] „What is CAN Bus?,“ [Online]. Available: https://canbuskits.com/what.php. [Přístup získán duben 2019].

[5] Š. Radek, „Vlastnosti a užití průmyslových sběrnic,“ [Online]. Available: http://www.elektrorevue.cz/clanky/04019/index.html. [Přístup získán duben 2019].

[6] „IEC 61158-1:2019,“ 10 duben 2019. [Online]. Available: https://webstore.iec.ch/publication/59890. [Přístup získán duben 2019].

[7] „Komunikační sběrnice PROFIBUS,“ [Online]. Available: https://coptkm.cz/portal/reposit.php?action=0&id=33311&revision=-1&instance=2. [Přístup získán duben 2019].

[8] „RS485/MODBUS-RTU ver. 3.0,“ [Online]. Available: https://www.rawet.cz/data/files/products/15-02-2017_19-50_protokol-mb.pdf. [Přístup získán duben 2019].

[9] „Modbus protocol,“ [Online]. Available: https://www.modbustools.com/modbus.html. [Přístup získán duben 2019].

[10] D. Ronald, Prosinec 2004. [Online]. Available: http://schusterusa.com/wp-content/uploads/2012/12/harting_industrial_ethernet_handbook.pdf. [Přístup získán duben 2019].

[11] O. Hynčica a F. Zezulka, „Průmyslový Ethernet,“ [Online]. Available: http://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-2005_04_30417_493/. [Přístup získán duben 2019].

[12] S. Barrie, Mistrovství - počítačové sítě, Brno: Computer Press, 2010.

[13] B. Madhav, „OSI Model Layers—“Explained”,“ 14 duben 2018. [Online]. Available: https://medium.com/@madhavbahl10/osi-model-layers-explained-ee1d43058c1f. [Přístup získán duben 2019].

[14] „The OSI Model - Features, Principles and Layers,“ [Online]. Available: https://www.studytonight.com/computer-networks/complete-osi-model. [Přístup získán duben 2019].

[15] „IP - Internetový protokol (Internet Protocol),“ [Online]. Available: https://managementmania.com/cs/internetovy-protokol-ip. [Přístup získán duben 2019].

[16] „Internet Protocol,“ [Online]. Available: https://www.cloudflare.com/learning/ddos/glossary/internet-protocol/. [Přístup získán duben 2019].

[17] „IP header,“ [Online]. Available: https://study-ccna.com/ip-header/. [Přístup získán duben 2019].

[18] „Transmission Control Protocol,“ [Online]. Available: https://www.extrahop.com/resources/protocols/tcp/. [Přístup získán duben 2019].

[19] J. Stretch, „Understanding TCP Sequence and Acknowledgment Numbers,“ 7 Červen 2010. [Online]. Available: http://packetlife.net/blog/2010/jun/7/understanding-tcp-sequence-acknowledgment-numbers/. [Přístup získán duben 2019].

[20] B. Mitchell, „TCP Headers and UDP Headers Explained,“ 22 duben 2019. [Online]. Available: https://www.lifewire.com/tcp-headers-and-udp-headers-explained-817970. [Přístup získán duben 2019].

[21] „TCP Flags,“ 4 říjen 2018. [Online]. Available: TCP Headers and UDP Headers Explained. [Přístup získán duben 2019].

[22] H. O. d. Beeck, „TCP Header,“ září 2002. [Online]. Available: http://telescript.denayer.wenk.be/~hcr/cn/idoceo/tcp_header.html. [Přístup získán duben 2019].

[23] „Computer Network | User Datagram Protocol (UDP),“ [Online]. Available: https://www.geeksforgeeks.org/computer-network-user-datagram-protocol-udp/. [Přístup získán duben 2019].

[24] „Common Indutrial Protocol,“ [Online]. Available: http://www.technologyuk.net/telecommunications/industrial-networks/cip.shtml. [Přístup získán duben 2019].

[25] M. Rostan, „Industrial Ethernet Technologies: Overview,“ únor 2014. [Online]. Available: https://www.ethercat.org/download/documents/Industrial_Ethernet_Technologies.pdf. [Přístup získán duben 2019].

[26] Siemens, „ProfiNet - Standard pro průmyslový Ethernet v automatizaci,“ duben 2005. [Online]. Available: http://stest1.etnetera.cz/ad/current/content/data_files/automatizacni_systemy/prumyslova_komunikace/profinet/profinet_04_2005_cz.pdf?fbclid=IwAR3tuqjN8v685gbskWsVF9_bE7_wmzen2nYfNzPQ98SjOTrhif7ev3d288U. [Přístup získán duben 2019].

[27] „AoE - ADS over EtherCAT,“ [Online]. Available: https://infosys.beckhoff.com/english.php?content=../content/1033/el6695/1317828363.html&id=5681562584494671228. [Přístup získán duben 2019].

[28] A. Fogini, „EtherCAT Automation Protocol,“ [Online]. Available: http://automa.cz/cz/casopis-clanky/ethercat-automation-protocol-2017_02_0_9811/. [Přístup získán duben 2019].

[29] „Industrial Ethernet Technologies: Overview," únor 2014. [Online]. Available: https://www.ethercat.org/download/documents/Industrial_Ethernet_Technologies.pdf. [Přístup získán duben 2019].

[30] E. T. Group, „EtherCAT Diagnostics," listopad 2018. [Online]. Available: https://www.ethercat.org/download/documents/EtherCAT_Diagnosis_For_Users.pdf. [Přístup získán duben 2019].

[31] „S7 Communication (S7comm)," [Online]. Available: https://wiki.wireshark.org/S7comm. [Přístup získán duben 2019].

[32] „About Modbus | Simply Modbus Software," [Online]. Available: http://www.simplymodbus.ca/faq.htm#Ext. [Přístup získán duben 2019].

[33] „MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE," 24 říjen 2006. [Online]. Available: http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf. [Přístup získán duben 2019].

[34] „Propojení PC a PLC pomocí Ethernet Powerlink," 2010. [Online]. Available: https://core.ac.uk/download/pdf/30280528.pdf. [Přístup získán duben 2019].

[35] F. Zezulka a O. Hynčica, „Průmyslový Ethernet IX: EtherNet/IP, EtherCAT," [Online]. Available: http://automa.cz/Aton/FileRepository/pdf_articles/37910.pdf. [Přístup získán duben 2019].

[36] „Ethernet/IP Communication," [Online]. Available: https://www.ethernet-powerlink.org/powerlink/industrial-ethernet-facts/selection-of-systems-for-review/ethernetip-communication. [Přístup získán duben 2019].

[37] J. Rinaldi, „Ethernet/IP Overview," [Online]. Available: https://www.rtautomation.com/technologies/ethernetip/. [Přístup získán duben 2019].

[38] „Co je OPC? OPC server? OPC klient?," 7 srpen 2013. [Online]. Available: https://www.foxon.cz/blog/prakticka-teorie/159-co-je-opc-opc-server-opc-klient. [Přístup získán duben 2019].

[39] „Řídicí jednotky IO-Link," [Online]. Available: https://www.pepperl-fuchs.com/czech_republic/cs/classid_6436.htm. [Přístup získán duben 2019].

[40] „IO-Link - řešení pro komunikaci v průmyslové automatizaci," únor 2008. [Online]. Available: http://automa.cz/cz/casopis-clanky/io-link-reseni-pro-komunikaci-v-prumyslove-automatizaci-2008_02_36687_5130/. [Přístup získán duben 2019].

[41] „IO-Link," [Online]. Available: https://www.rtautomation.com/technologies/io-link/. [Přístup získán duben 2019].

[42] A. Vojáček, „Programovací režimy pro PLC dle IEC 61131-3 (CoDeSys)," 3 březen 2011. [Online]. Available: https://automatizace.hw.cz/programovaci-rezimy-pro-plc-dle-iec-611313-codesys. [Přístup získán duben 2019].

[43] „CODESYS - the comprehensive software suite for automation technology," [Online]. Available: https://www.codesys.com/the-system.html. [Přístup získán duben 2019].

[44] R. P. Foundation, „What is a Raspberry Pi?,“ [Online]. Available: https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/. [Přístup získán duben 2019].

[45] „Chapter 1. Introduction,“ [Online]. Available: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html. [Přístup získán duben 2019].

[46] „WinPcap,“ [Online]. Available: https://www.winpcap.org/. [Přístup získán duben 2019].

[47] „SampleCaptures,“ [Online]. Available: https://wiki.wireshark.org/SampleCaptures. [Přístup získán duben 2019].

[48] G. Molenaar, „Welcome to python-snap7's documentation!,“ 2019. [Online]. Available: https://python-snap7.readthedocs.io/en/latest/. [Přístup získán duben 2019].

[49] „The standard library for modbus communication,“ 2017. [Online]. Available: http://easymodbustcp.net/en/. [Přístup získán duben 2019].

[50] „Simple Open EtherCAT Master or SOEM,“ [Online]. Available: https://openethercatsociety.github.io/doc/soem/. [Přístup získán duben 2019].