VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2017 Bc. Pavel Marek

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

IMPLEMENTACE CHYTRÉ TOVÁRNY
IMPLEMENTATION OF SMART FACTORY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Pavel Marek

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. Vladislav Škorpil, CSc.

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

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

Student: Bc. Pavel Marek ***ID:*** 136558
Ročník: 2 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Implementace chytré továrny

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte problematiku plošného sběru dat v průmyslu a možnosti komunikace s průmyslovými zařízeními řízenými PLC (Programmable Logic Controller). Navrhněte a realizujte komunikační síť založenou na standardu Ethernet nebo nějaké jeho průmyslové verzi. Účelem této sítě bude centralizovaný sběr dat z průmyslových měřicích zařízení do databáze v nadřazeném systému. Měřicí zařízení budou nadřazenému systému poskytovat především výsledky měření, identifikaci měřených výrobků, zvolené měřicí metody a informace o aktuální vytíženosti. Zároveň budou měřicí zařízení, s ohledem na právě měřený výrobek, vybírat z databáze nadřazeného systému příslušné měřicí metody s ohledem na plynulý a bezpečný chod. Vytvořte aplikaci pro nadřazený systém, která bude získaná data zpracovávat a přehledně zobrazovat a ve které bude také možnost definovat měřené výrobky pomocí měřicích metod. Vše realizujte s ohledem na koncept Průmysl 4.0.

DOPORUČENÁ LITERATURA:

[1] ZURAWSKI, R. Industrial communication technology handbook. Second edition. Boca Raton, CRC Press, Taylor & Francis Group, 2015. ISBN 9781482207323.

[2] AGARWAL, V.; HUDDLESTON, J. Databáze v C# 2008: průvodce programátora. Computer Press, Brno, 2009. ISBN 978-80-251-2309-6.

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 24.5.2017

Vedoucí práce: doc. Ing. Vladislav Škorpil, CSc.
Konzultant: Ing. Petr Povolný

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

ABSTRAKT

Tato diplomová práce se zabývá studií konceptů Internetu věcí, Průmyslu 4.0 a současného stavu v továrnách. Na základě provedených studií byla navržena a realizována úprava hardwaru a softwaru průmyslových strojů a realizace připojení strojů do komunikační sítě. Byla také navržena a naprogramována aplikace v jazyce C#. Tato aplikace zajišťuje centralizovaný sběr dat z průmyslových strojů, poskytuje strojům a lidem v továrně různé služby a zpracovává a zobrazuje potřebná data. K těmto účelům využívá aplikace databázový systém založeny na SQLite. Tyto úkony návrhu a realizace jsou shrnuty do systému určeného pro implementaci do chytré továrny, který odpovídá konceptu Průmyslu 4.0.

KLÍČOVÁ SLOVA

Internet věcí, průmyslový Internet věcí, Průmysl 4.0, chytrá továrna, C#, PLC, OPC server, OPC klient, databázové systémy, komunikační technologie, Ethernet, Profinet, TCP/IP, aplikace pro zpracování a zobrazování dat

ABSTRACT

This diploma thesis is focused on the study of concept of Internet of Things, concept of Industry 4.0 and on current conditions in smart factories. Based on these studies there was designed and implemented hardware and software adjustment for industrial machines and connection of these industrial machines to communication network. There was designed and programmed the application in the C# language. This application provides a data collection from industrial machines, provides various services for machines and humans and this application is processing and viewing necessary data. For these purposes the application is using a database system based on the SQLite. These tasks of designing and implementation are summarized to system, which is determined for smart factory implementation. This implementation is created according to Industry 4.0 concept.

KEYWORDS

Internet of Things, Industrial Internet of Things, Industry 4.0, smart factory, C#, PLC, OPC server, OPC client, database systems, communication technologies, Ethernet, Profinet, TCP/IP, data processing and viewing application

MAREK, Pavel *Implementace chytré továrny*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 80 s. Vedoucí práce byl doc. Ing. Vladislav Škorpil, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Implementace chytré továrny“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Ing. Vladislavu Škorpilovi, CSc. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Technicka 12, CZ-61600 Brno, Czechia

http://www.six.feec.vutbr.cz

Výzkum popsaný v této diplomové práci byl realizovaný v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

OBSAH

SEZNAM OBRÁZKŮ

1 ÚVOD

Díky vzrůstající ekonomice se závratným tempem zvyšuje i tempo výroby a produkce průmyslových továren. S tím souvisí i implementace stále nových strojů, které jsou určeny k pokrytí této zvýšené potřeby výroby. Nové stroje jsou zároveň koncipovány tak, aby byly schopny pracovat samostatně s co nejmenším zásahem obsluhy. Právě díky tomu vznikají stále častěji požadavky managementu továren a také jejich zákazníků na sběr komplexních dat z jednotlivých strojů. Data jsou nositeli informací o vykonávané výrobě a o vyráběných produktech. Získávání těchto dat je následně využíváno pro zlepšení kvality výroby. Jejich analýza je zároveň využívána pro zlepšení činností výrobních strojů a vyhodnocování jejich kondice. S rozvojem komunikačních technologií vzniká i mnoho prostředků, jak tyto požadavky splnit. Tato skutečnost umožňuje postupnou implementaci Internetu věcí do průmyslových továren. Právě s tímto cílem implementace Internetu věcí do průmyslu vznikl koncept Průmysl 4.0, který má vytvářet tzv. chytré továrny s chytrou výrobou.

Implementace Průmyslu 4.0 do továren však není jen pro nová zařízení. Původní stroje vykonávají svou práci stejně kvalitně. Proto jejich transformace na zařízení určená pro chytrou továrnu také nemůže být opomíjena. Sběr a analýza dat z těchto strojů je pro management a cílového zákazníka stejně důležitá, jako je tomu i u nových zařízení. Právě přizpůsobení současných strojů do konceptu Průmyslu 4.0 s využitím současných komunikačních technologií a technologií pro sběr dat se věnuje tato diplomová práce. Cílem je tedy vytvoření sítě, ve které bude prováděn centralizovaný sběr dat a zajištěna jejich prezentace a jednoduchá analýza. Stejně důležitým cílem je také návrh a realizace transformace strojů běžné továrny na systémy vhodné pro chytrou továrnu.

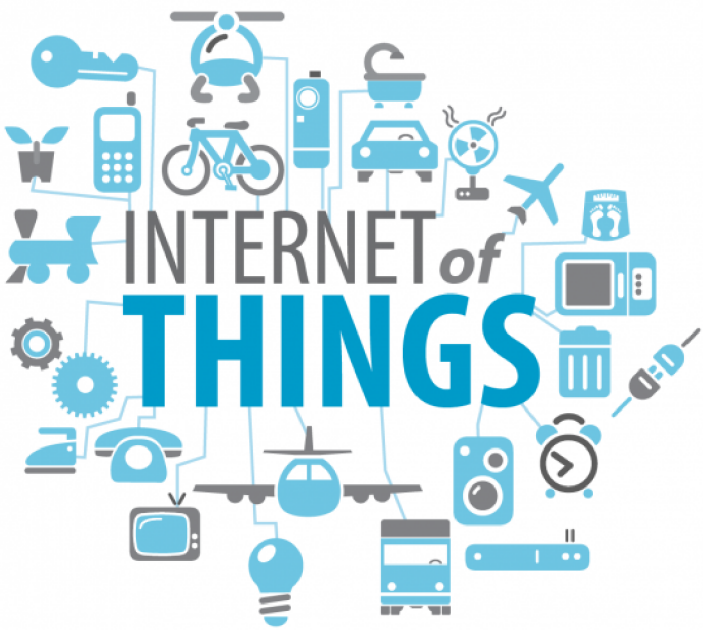
Z těchto důvodů je v této práci nejprve prostudován obecně koncept Internetu věcí a jeho technické myšlenky a specifikace. Navazuje studium implementace Internetu věcí v průmyslovém prostředí. Z tohoto důvodu je popsán koncept Průmyslu 4.0 a jeho vliv na chování průmyslových strojů a průmyslových aplikací. Tento koncept následně tvoří manuál pro implementaci systému pro chytrou továrnu. Teoretická část je zakončena rozborem současných možností síťové komunikace s jednotlivými stroji a rozborem možností sběru velkoobjemových dat z průmyslových strojů. Studium využitelných komunikačních technologií a technologií sběru dat je zaměřeno na jejich využití a aplikaci v průmyslu. Poznatky získané z teoretického rozboru jsou nejprve využity při náhledu na současný stav v továrnách a následně jsou použity pro návrh a implementaci systému určeného pro chytré továrny.

2 INTERNET VĚCÍ

Implementace chytré továrny a aplikování Průmyslu 4.0 jsou založeny na propojení hardwarových a softwarových komponent tvořících automatizační systém v továrně. Toto propojení a následné připojení těchto komponent k Internetu se označuje názvem Internet věcí (v originále Internet of Things, zkratka IoT). Právě na Internetu věcí je koncept Průmyslu 4.0 založen, proto budou Internet věcí a jeho principy v této kapitole podrobněji představeny.

2.1 Cíle Internetu věcí

Internet věcí je označení pro koncept, který představuje prostředí obsahující širokou škálu objektů neboli „věcí“. Tyto objekty jsou propojeny pomocí drátové nebo bezdrátové sítě a jsou jim přiřazeny adresy podle příslušného schématu adresování. Objekty tedy tvoří síť, která jim umožňuje společnou komunikaci a také umožňuje reagovat na vykonávané úkony ostatních objektů. Vzájemné propojení umožňuje vytvoření nových služeb a napomáhá jednotlivým objektům k dosažení společných cílů. Tento koncept vytváří inteligentní prostředí, ve kterém je umožněno „věci“ se kdykoliv a kdekoliv spojit s kýmkoliv nebo s čímkoliv s využitím jakékoliv dostupné síťové cesty či služby. [1]



Obr. 2.1: Myšlenka Internetu věcí [2].

Internet věcí lze považovat za revoluci klasického Internetu. Připojené objekty se zde navzájem rozpoznávají. Mají schopnost dělat vlastní rozhodnutí a možnost ovlivňovat ostatní objekty na základě informací, které poskytují nebo naopak přijímají. To z nich vytváří inteligentní účastníky síťové komunikace. Objekty mají tedy přístup k informacím ostatních objektů a mohou být součástí komplexních náročných služeb. Takto definované schéma představuje základ pro využívání tzv. „cloud computing“[1]. Účastníci zde mohou přistupovat k nabízeným centralizovaným službám a využívat je ke zpracovávání dat a vykonávání vlastních úkolů. [1]

V současné době vznikají nové typy aplikací, například automaticky řízený elektromobil nebo chytrá domácnost, kde jednotlivé prvky těchto aplikací poskytující různé služby. Jedná se například o oznamování různých událostí, bezpečnost, úsporu energie, telekomunikace, automatizaci úkonů nebo počítač. To je základem pro jednoduchou síť s definovaným uživatelským rozhraním. Člověk pak může k těmto službám přistupovat pomocí rozhraní z počítače z práce nebo chytrého telefonu. Právě lidé, chytré objekty, stroje a další platformy (například bezdrátové či drátové senzory, IP kamery) zaplňující komunikační prostor tvoří decentralizované zdroje služeb. Tyto služby jsou navzájem propojeny adaptivní sítí, která se však sama o sobě také skládá z několika dalších sítí. Jednotlivé sítě pak mají definované způsoby komunikace, které popisují různé komunikační protokoly. V tomto kontextu je právě Internet věcí obecný název pro tuto síť sítí, ve které je propojením různých zařízení vytvořeno chytré prostředí. A právě takto vytvořené chytré prostředí poskytuje přístup k novým informacím a službám. Není důležité, zda je k těmto účelům využito drátové nebo bezdrátové komunikace. [1, 3]

Internet věcí vytváří spolupráci Internetu průmyslového (například průmyslové stroje, firemní síť), klasického spotřebitelského (weby pro volný čas) a obchodního (například internetový vyhledávač Google, internetové bankovnictví). Toto spojení vytváří bránu pro vzájemné spojení lidí, dat a strojů. [1]

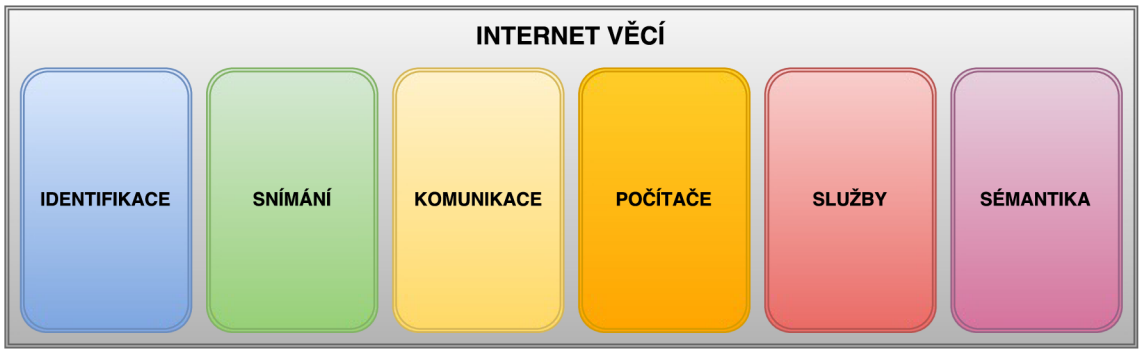
2.2 Technologie Internetu věcí

V této podkapitole je popsáno 6 hlavních součástí, které zajišťují funkčnost Internetu věcí [5]. Jedná se o identifikaci, snímání, komunikaci, výpočty, služby a sémantiku. Těchto 6 součástí tvoří internet věcí jako celek, znázorněno na obrázku číslo 2.2.

2.2.1 Identifikace

Pro vzájemnou komunikaci je nutné každé zařízení, které se chce připojit do sítě, jednoznačně identifikovat. Identifikace složí k pojmenování objektů a přiřazení pří-

[1]Poskytování serverových služeb a aplikací, které jsou dostupné prostřednictvím sítě.



Obr. 2.2: Internet věcí.

slušných služeb. Pro identifikaci lze využít například metody elektronické identifikace produktu (EPC - Electronic Product Code) nebo všudypřítomných kódů (uCode - Ubiquitous Code). Kromě názvu je nutné zařízení také přiřadit jednoznačnou dresu. Protože je Internet věcí součástí globální sítě Internet, je pro účely adresování využíván protokol IP (Internet Protocol). Proto je pozornost zaměřena na zavedení standardu IPv6, který zprostředkovává dostatečně velký prostor využitelných adres pro Internet věcí. IPv6 definuje 128 bitové adresy a umožňuje adresaci 340×10^{36} zařízení [4]. To představuje zdánlivě nevyčerpatelný zdroj adres. Pokud však budou zařízení přibývat nevídanou rychlostí, bude nutné do značné míry upravit správu rozsahu IPv6. Tato skutečnost představuje velkou nevýhodu. Další nevýhodou je, že také v mnoha dnešních zařízeních určených pro připojení do Internetu věcí není možné využít IPv6, ale jsou koncipovány pro IPv4 [3]. Avšak i adresní schéma IPv4 umožňuje připojení do Internetu věcí. [5]

Po zajištění identifikace objektu je nutné zajistit, aby se ostatní objekty dozvěděly o datech a o službách, které nově připojený objekt poskytuje. Objekt zároveň musí poskytovaná data uzpůsobit požadavkům na zabezpečení. [1]

2.2.2 Snímání

Snímání představuje sběr dat a odesílání získaných dat do příslušného úložiště, databáze nebo cloudu. Snímání zajišťují chytré senzory a snímače různých veličin nebo přenositelná chytrá zařízení. Mezi tato zařízení patří jednoduché počítače s vestavěným senzorem, TCP/IP (Transmission Control Protocol/Internet Protocol) rozhraním a akcelerátorem pro zabezpečení. Typickými představiteli těchto zařízejní jsou Arduino a Raspberry PI. [5]

2.2.3 Komunikace

Komunikační technologie Internetu věcí umožňují vzájemně propojit různorodá zařízení. Z tohoto důvodu Internet věcí počítá s mnoha komunikačními technologiemi [5]. Zde je uveden přehled několika z nich:

- **Průmyslový Ethernet** (Určený pro průmyslové prostředí, umožňuje řízení v reálném čase. Existují různé verze: EtherCAT, EtherNet/IP, PROFINET, POWERLINK, CC-Link IE a Modbus/TCP).
- **WiFi** (Označení stadardů IEEE 802.11, které definují bezdrátovou komunikaci.)
- **IEEE 802.15.4** (Standard definující rádiové sítě krátkého dosahu - WPAN. Určen pro průmyslové využití.)
- **ZigBee** (Technologie založená na standardu IEEE 802.15.4.)
- **Z-Wawe** (Protokol pro automatizované řízení objektů v chytré domácnosti.)
- **RFID** (Radio Frequency Identificator - Pro radiovou identifikaci různých objektů.)
- **NFC** (Near Field Communication - Technologie určená pro komunikaci objektů na velmi krátkou vzdálenost, do 4 centimetrů.)
- **UWB** (Ultra Wide Band - Bezdrátová technologie pracující s nízkou úrovní signálu určená pro komunikaci senzorů [5].)
- **LTE, LTE-A** (Long-Term Evolution - Standard bezdrátové komunikace mobilních zařízení. LTE Advanced - Rozšíření šířky pásma technologie Long-Term Evolution pro získání lepších vlastností přenosu.)
- **Bluetooth Low Energy** (Standard pro bezdrátovou komunikaci chytrých objektů, komunikace má malé nároky na spotřebu energie.)
- **SIGFOX** (Bezdrátová síť založená na malé spotřebě energie. Je určená přímo pro Internet věcí. [5])
- **WiMAX** (Pro bezdrátovou distribuci dat ve venkovní síti - na větší vzdálenosti.)
- **HomePlug** (Využívá elektrické rozvody 230V pro přenos dat.)
- **HomePNA** (Pro vysokorychlostní datový přenos po vnitřních telefonních rozvodech.)
- **HomeGrid/G.hn** (Standard pro vysokorychlostní datový přenos po koaxiálních kabelech, telefonních rozvodech a elektrické síti 230V.)
- **LonWorks** (Průmyslová komunikační sběrnice. Určené pro automatizaci, měření a regulace.)

2.2.4 Počítače

Počítače vykonávající výpočty a zajišťující chod softwarových aplikací tvoří mozek Internetu věcí. Do této skupiny patří různé hardwarové platformy, například Arduino, Intel nebo Raspberry PI. [5]

Pro zajištění funkcí Internetu věcí se využívají operační systémy reálného času, které pracují na těchto platformách. Operační systéme reálného času umožňují okamžitou reakci na aktuální informace a okamžitý zásah. Příklady operačních systémů reálného času: Contiki, TinyOS nebo FreeRTOS. [5]

2.2.5 Služby

Služby lze rozdělit do 4 kategorií. První kategorií je služba související s identifikací (*Identify-Related Services*). Každá aplikace, která potřebuje přenést objekt z reálného světa do světa virtuálního, musí objekt identifikovat. Tuto službu vyžaduje mnoho dalších služeb. [5]

Druhou kategorií jsou shromažďovací funkce (*Information Aggregation Services*). Shromažďovací funkce sbírá a vyhodnocuje data ze senzorů, které potřebují být zpracovány v Internetu věcí. [5]

Třetí kategorie jsou služby spolupráce (*Collaboration-Aware Services*), které využívají nahromaděná data k vykonávaní rozhodovacích procesů a k příslušným reakcím na tato data. [5]

Poslední kategorii představují takzvané všudypřítomné služby (*Ubiquitous Services*). Jejich hlavním cílem je poskytovat služby spolupráce (3. kategorie služeb) kdykoliv, kdy jsou vyžadovány, komukoliv, kdo je potřebuje a kdekoliv, kde jsou vyžadovány. [5]

2.2.6 Sémantika

Výraz sémantika vyjadřuje? v souvislosti s Internetem věcí, schopnost získat elegantně znalosti různých strojů za účelem poskytování a zkvalitňování vyžadovaných služeb. Sémantika reprezentuje „schopnost myšlení“ v Internetu věcí nad odesíláním požadavků, aby bylo dosaženo nejvhodnějších požadovaných služeb. [5]

2.3 Architektura Internetu věcí

Architekturu Internetu věcí lze nejvhodněji popsat rozdělením do 5 vrstvového modelu [5]. Jedná se o vrstvu objektů, abstraktní vrstvu, řízení služeb, aplikační vrstvu a vrstvu dohledu. Architektura je znázorněna na obrázku číslo 2.3 a úkoly jednotlivých vrstev jsou pod obrázkem blíže specifikovány.

Obr. 2.3: Architektura Internetu věcí.

2.3.1 Vrstva objektů

Tato nejnižší vrstva modelu reprezentuje fyzická zařízení a technologie připojení do Internetu věcí. Na této vrstvě jsou využívány algoritmy pro automatické přidávání objektů do Internetu věcí. Na této vrstvě vznikají velkoobjemová data Internetu věcí. [5]

Připojená zařízení zajišťují měření (senzory) dat nebo vykonávají své úkony (akční členy). Změřená data nebo informace o svých zásazích předávají pomocí bezpečných kanálů vrstvě virtualizace. [5]

2.3.2 Abstraktní vrstva

Úkolem této vrstvy je doručit data z vrstvy objektů do vrstvy řízení služeb. Pro přenos dat lze využít různých technologií: RFID, 3G, GSM, UMTS, WiFi, infračervený přenos, Bluetooth a další podobné. Na této vrstvě je realizováno zpracování dat a „cloud computing“. [5, 6]

2.3.3 Vrstva řízení služeb

Na této vrstvě dochází k přiřazení požadované služby k žadateli, který je definován adresou. Tato vrstva umožňuje softwarovou práci s různorodými objekty. K práci s nimi není nutná znalost hardwarových platforem. [5, 6]

Tato vrstva také zpracovává přijímaná data a na základě vlastních rozhodnutí jim zajišťuje služby s využitím dostupných síťových protokolů. [5]

2.3.4 Aplikační vrstva

Aplikační vrstva zajišťuje služby, které si vyžádal zákazník. Například prezentuje zákazníkovi data o teplotě a vlhkosti vzduchu, které si vyžádal. Nejdůležitějším úkolem aplikační vrstvy je poskytování chytrých a kvalitních služeb a pokrýt všechny zákazníkovy požadavky. [5, 6]

Tato vrstva reprezentuje například chytrou domácnost, chytrou zdravotní péči nebo průmyslovou automatizaci. jedná se o rozhraní pro zákazníka. [5]

2.3.5 Vrstva dohledu

Na této nejvyšší vrstvě dochází ke správě systémových akcí a služeb v celém Internetu věcí. Cílem této vrstvy je vytvářet obchodní model, grafy, analýzu, vyhodnocování a monitoring dat přijatých od aplikační vrstvy. také má za úkol rozvíjet jednotlivých částí v Internetu věcí. Vykonává sledování a řízení všech čtyř předchozích vrstev. [5, 6]

Pro zlepšení služeb a udržování bezpečnosti na této vrstvě dochází k porovnávání výstupů z jednotlivých vrstev s očekávanými výstupy. Na základě výsledků těchto porovnávání se pak provádí interakce s ostatními vrstvami. [5]

2.4 Komunikační modely

Komunikační model udává, jakým způsobem si mohou jednotlivá síťová zařízení vyměňovat data a jak využívat informace novými způsoby. Také Internet věcí má stanovené komunikační modely, které jsou popsány v dokumentu RFC 7452, [7]. Právě popisu těchto topologií se tato podkapitola věnuje.

2.4.1 Model Device-to-Device

V tomto komunikačním modelu jsou dvě nebo více zařízení vzájemně přímo propojena a komunikují přímou cestou. Pro realizaci spojení může být využito poměrně

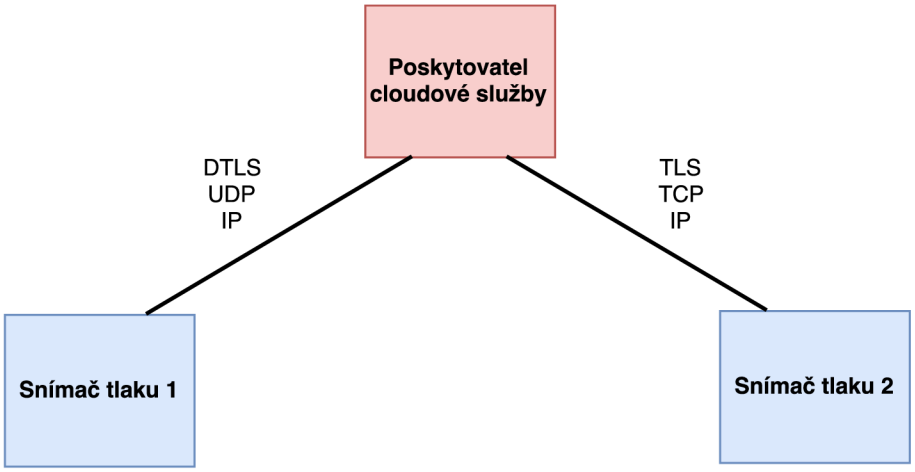
velké množství komunikačních protokolů: IP sítě, Internet, častěji však Bluetooth, ZigBee, atd. Topologie je znázorněna na obrázku č. 2.4. [3, 7]

Obr. 2.4: Znázornění komunikačního modelu Device-to-Device.

Tato topologie umožňuje připojeným zařízením vzájemnou výměnu příkazů a dat, na jejichž základě provádějí požadované úkony. Tato topologie je vhodná pro chytrou domácnost, kde se přenášené informace posílají v malých datových jednotkách. Zařízení určené pro tyto účely často disponují relativně nízkými nároky na přenosovou kapacitu. [3, 7]

2.4.2 Model Device-to-Cloud

V této topologii je zařízení připojeno pomocí Internetu do cloudové služby. Službou může být aplikace poskytující výměnu dat a kontrolu datového provozu. Výhodou je, že k připojení lze použít tradičního drátového Ethernetu nebo bezdrátové WiFi k vytvoření spojení mezi zařízením a IP sítí, která zajišťuje přístup ke cloudovým službám. Topologie je znázorněna na obrázku č. 2.5. [3, 7]

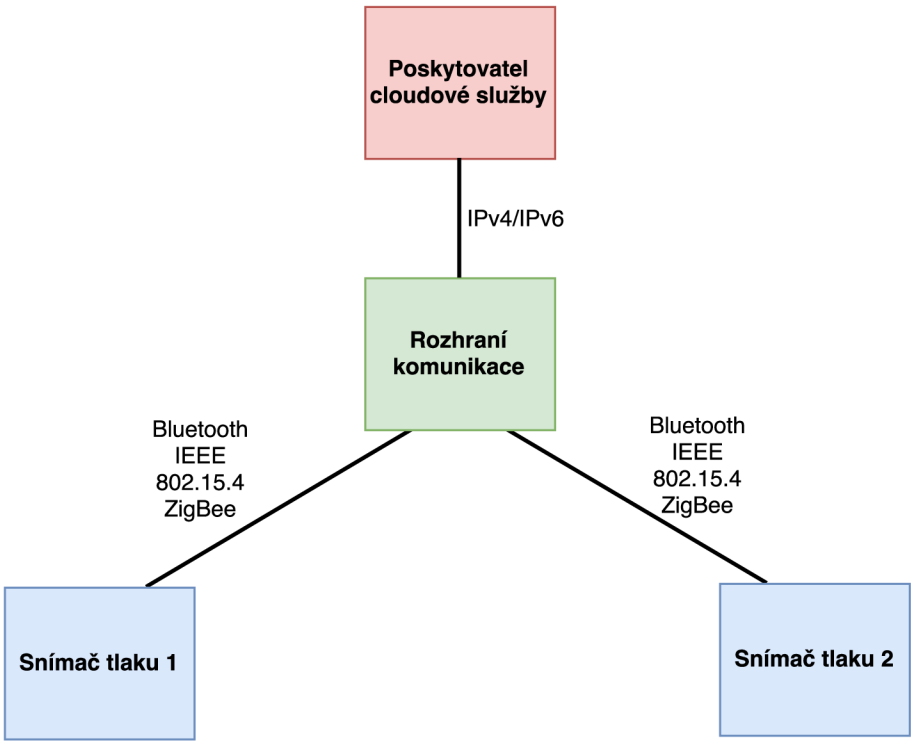


Obr. 2.5: Znázornění komunikačního modelu Device-to-Cloud.

Připojené zařízení (například snímač teploty vytápění) zaznamenává teploty a odesílá je do cloudu ke zpracování. Zde se z přijatých údajů počítá statistika a lze určit například spotřebu energie. [3, 7]

2.4.3 Model Device-to-Gateway

Komunikační model Device-to-Cloud je primárně určen pro zařízení, které jsou ke komunikaci schopny využívat široce rozšířené stadardy Ethernet nebo WiFi. Existuje však mnoho dalších vestavěných zařízení, které jsou založeny na aplikacích, které stále nepatří mezi příliš rozšířené. Mezi tyto aplikace patří například IEEE 802.15.4 nebo speciální funkce pracující na aplikační vrstvě (například aplikace pro ověření pravosti a oprávnění), které nevyužívají protokolů založených na IP síti. Z těchto důvodů vznikl požadavek na vytvoření topologie s rozhraním, které dokáže vzájemně propojit aplikace různých technologií a standardů. Rozhraní zde figuruje jako překladač mezi jednotlivými komunikačními protokoly. Rozhraní také může zajistit překlad adres zařízení s různými adresními schématy na IPv4 nebo IPv6. Topologie je znázorněna na obrázku č. 2.6. [3, 7]

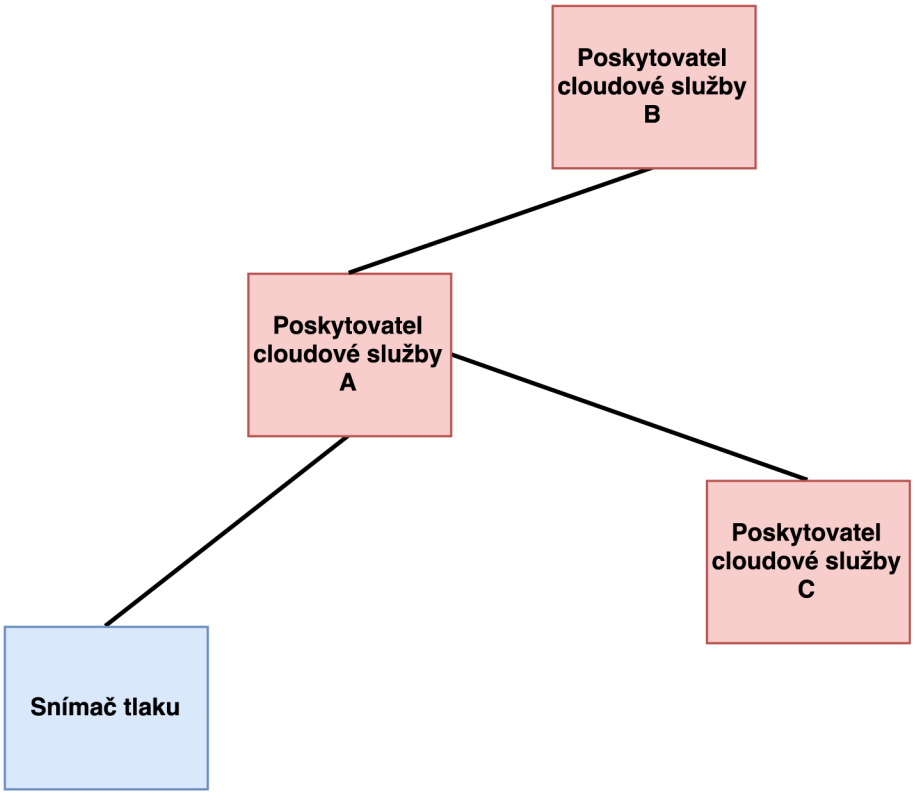


Obr. 2.6: Znázornění komunikačního modelu Device-to-Gateway.

Cílem topologie je tedy připojit zařízení různých výrobců do Internetu věcí a odstranit případné problémy s vzájemnou nekompatibilitou. [3, 7]

2.4.4 Model Back-End Data-Sharing

Komunikační model Device-to-Cloud poskytuje komunikaci pouze s poskytovatelem cloudové služby a má přístup pouze do svého úložiště. Objekt připojený do Internetu věcí však potřebuje mít schopnost využívání cloudové služby s kombinací schopnosti zpracování dat, které mu poskytují ostatní objekty. Proto je nutné zajistit přístup do třetí služby, kde budou k dispozici data od dalších účastníků připojených do Internetu věcí. Právě model Back-End Data-Sharing vytváří architekturu, která nabízí objektu možnost odesílat data cloudové službě a analyzovat je v kombinaci právě s daty ze všech ostatních zdrojů. Topologie je znázorněna na obrázku č. 2.7. [3, 7]



Obr. 2.7: Znázornění komunikačního modelu Back-End Data-Sharing.

Vytvořením modelu, kde datová komunikace zahrnuje cloudy, uživatele a poskytovatele služeb zajišťuje snadný sběr dat, analýzu dat, vizualizaci dat a předvídavou analytickou technologii. Právě tyto součásti lze využít pro získání komplexních informací v Internetu věcí. [3, 7]

2.5 Aplikace Internetu věcí

Internet věcí lze aplikovat prakticky do všech odvětví dnešního světa, dělá tato odvětví přístupnější a usnadňuje jejich využívání. V této podkapitole je uveden rozsáhlý přehled odvětví, které internet věcí ovlivňuje nebo v blízké budoucnosti ovlivňovat bude [1]. Z uvedeného přehledu níže vyplývá, že Internet věcí již ovlivňuje nebo v budoucnosti ovlivní téměř vše.

Chytrá města

- Chytré parkování: Monitorování dostupnosti volných míst ve městech.
- Kondice materiálů: Monitorování vibrací a stavu materiálů různých staveb.
- Sledování elektromagnetického pole: Měření energie vyzařované bezdrátovými zařízeními.
- Sledování dopravy: Monitorováni vozidel a určování vytíženosti komunikací s cílem optimalizovat dopravu.
- Chytré osvětlení: Řízení osvětlení města v závislosti na denní době a aktuálním počasí.
- Chytré silnice: Inteligentní silnice, které budou schopny uživatele varovat v závislosti na provozu, dopravních nehodách a počasí.

Chytré životní prostředí

- Sledování lesů a přírody: Monitorování rizik vzniku požárů lesů a zatravněných prostorů a dělání preventivních opatření.
- Sledování čistoty vzduchu: Monitorování koncentrace CO2 ve vzduchu u továren a u dalších kritických oblastí.
- Sledování úrovně sněhu: Měření úrovně a kvality sněhu v reálném čase a analýza rizika lavin.
- Sledování rizik sesuvu půdy: Analýza vlhkosti půdy a měření vibrací s cílem odhalit potencionální nebezpečí.
- Sledování pohybu tektonických desek a detekce zemětřesení: Zajištění všech nutných opatření na místech zemětřesení.

Chytrá hospodaření s vodou

- Sledování pitné vody: Měření kvality pitné vody v obydlených oblastech.
- Sledování chemických látek ve vodě: Měření úniku chemikálií a odpadů do vod.
- Sledování kvality vody v bazénech: Měření parametrů vody v plaveckých bazénech.
- Sledování znečistění moří: Kontrola znečištění moří v reálném čase.
- Sledování ztrát vody: Detekce vody v okolí zásobníků vody a potrubí pro od-

halení vzniklých děr a netěsností.
- Sledování hladiny vod: Měření hladiny vody v řekách nebo jezerech pro detekci možnosti záplav.

Chytrá měření
- Chytrá rozvodná síť elektřiny: Měření a řízení spotřeby energie v reálném čase.
- Měření zásobníků: Monitorování olejů, paliv a plynů ve skladech a přepravních cisternách.
- Chytrá fotovoltaická síť: Monitorování a optimalizace výkonu solární energie.
- Měření tlaku vody: Monitorování tlaku vody v distribučním systému.
- Sledování zásob: Monitorování nutnosti doplnění docházejících zásob nejrůznějšího zboží.

Chytrá bezpečnost a ochrana
- Řízení přístupu a detekce osob: Slouží k monitorování pohybu osob v prostorách, do kterých nemají mít přístup.
- Měření vlhkosti: Monitorování vlhkosti a přítomnosti vody v prostorech, ve kterých může vlhkost a voda způsobit nějaké škody.
- Měření radiace: Monitorování úrovně radiace v oblastech zvýšeného rizika, například v okolí jaderných elektráren.
- Sledování přítomnosti výbušných a hořlavých plynů: Monitorování úrovně nebezpečných plynů v průmyslovém prostředí.

Chytré obchodování
- Sledování zboží: Monitorování podmínek potravin během transportu.
- Chytré platby: Zjednodušení plateb za zábavu a potřeby, například s využitím mobilního telefonu.

Chytrá logistika
- Sledování kvality přepravy: Monitorování kritických podmínek při přepravě jako jsou vibrace, neoprávněné vybalování nebo škodící teplota.
- Lokalizace zboží: Snadné hledání zboží na velkých prostorech nebo ve velkých skladech.
- Sledování dopravních tras: Monitorování a řízení přepravy citlivého zboží, jako jsou drogy, léky, zbraně a jiné nebezpečné položky.

Chytré zemědělství
- Sledování počasí: Monitorování podmínek počasí pro vytváření předpovědí.

- Sledování stavu kompostu: Monitorování a řízení podmínek pro vypěstování kompostu.
- Sledování zvěře: Dohled nad vývojem mláďat pěstovaného dobytka pro zajištění optimálních podmínek pro přežití a zdraví.

Chytré domácnosti a automatizace domácností

- Sledování spotřeby energií a vody: Vyhodnocení spotřeb a analýza pro dosažení úspory jak výdajů, tak i energií.
- Vzdálená správa: Možnost ovládat nemovitost vzdáleně pro zajištění komfortu a předcházení nebezpečných situací.
- Zabezpečení nemovitostí: Detekce stavu oken a dveří a technologie protipožárních ochran a ochrany proti neoprávněnému vniknutí.
- Sledování podmínek skladování: Monitorování podmínek pro skladování potravin v restauracích nebo skladování cenných předmětů v muzeích.

Chytré zdravotnictví

- Sledování osob: Monitorování osamělých nemocných nebo starších osob a poskytnutí okamžité pomoci v případě potřeby.
- Sledování sportovců: Monitorování a optimalizace výkonu sportovců v reálném čase.
- Sledování pacientů: Monitorování aktuálního zdravotního stavu pacientů v nemocnicích.
- Sledování UV záření: Měření úrovně ultrafialového záření ze Slunce v reálném čase a poskytování okamžitého upozornění osobám.

Chytrý průmysl a automatizace průmyslové výroby

- Budování chytrých továren.
- Využití Internetu věcí pro realizaci konceptu Průmysl 4.0.
- Tomuto odvětví jsou věnovány následující kapitoly této práce.

2.6 Bezpečnost v Internetu věcí

Internet věcí musí být navržen tak, aby bylo možné snadno zrealizovat zabezpečení, které nebude zatěžovat výpočetní prostředky a bude schopno pracovat s velkým množstvím zařízení (řádově miliardy). Uživatelé, kteří budou využívat Internet věcí, potřebují mít jistotu bezpečnosti a spolehlivosti dat a údajů, které se na Internetu věcí pohybují. Aby mohli objekty bezpečně komunikovat s jinými objekty, je nutné zajistit důvěryhodnost, autentičnost (pravost) a integritu sdílených infor-

mací. Totéž platí při komunikaci mezi člověkem a věcí/strojem. Zde je nutné předejít neautorizované vniknutí nebo sledování a odposlechu dat. [1, 3, 7]

Cílem zabezpečení je tedy vytvoření spolehlivého prostředí, které bude s lidmi a objekty pracovat jako s uživateli (chrání osobní údaje lidí a pro oprávněné objekty představuje robustní systém se zajištěním stálé dispozice služeb). Spolehlivé prostředí musí pokrýt tyto požadavky:

- Řízení přístupu pro ochranu dat (vytvoření databáze s popisem, k jakým datům a jak s nimi může každý objekt v síti nakládat) [1].
- Metody pro zajištění důvěryhodnosti různých platforem (povolený hardware, software, komunikační protokoly, atd.) [1].
- Zajištění důvěryhodnosti informací (udržovat redundantní data, která budou sloužit k posouzení důvěryhodnosti informací získaných z Internetu věcí) [1].
- Vytvoření výpočetně nenáročné infrastruktury správy a distribuce veřejných klíčů (PKI - Public Key Infrastructure) jako základ pro ověření důvěryhodnosti [1].
- Případně decentralizovaný automaticky udržovaný systém pro ověření důvěryhodnosti objektů jako alternativa k PKI [1].
- Zajištění výpočetně nenáročného a důvěryhodného vytvoření šifrovaného spojení objektů a vyjednání parametrů šifrované komunikace s minimálním využitím komunikačních prostředků [1].

Kromě spolehlivého prostředí je nutné také zabezpečit vlastní data, protože i v Internetu věcí se může vyskytovat velké množství kritických údajů, které je nutné chránit proti neoprávněnému využití. Pro ochranu těchto dat je nutné splnit následující požadavky:

- Použití vhodných kryptografických metod k zašifrování dat, aby bylo zajištěno, že bez znalosti šifrovacího klíče nebude k datům přístup [1].
- Minimalizace pohybu citlivých dat v Internetu věcí [1].

Pro zajištění bezpečnosti v Internetu věcí lze ze současných technologií využít hardwarově akcelerované šifrování (AES - Advanced Encyrption Standard pro šifrování dat, ECC - Eliptic Curve Cryptography pro bezpečné vyměňování klíčů), zabezpečené sítě (VPN, SSH Tunnel) a širokou škálu firewallů.

3 PRŮMYSL 4.0

Internet věcí se stále více a více promítá i v průmyslové výrobě a automatizaci. Z tohoto důvodu vznikl koncept, který se nazývá Průmysl 4.0. Tento koncept je považován za čtvrtou průmyslovou revoluci. Jeho hlavním cílem je transformace současných továren a současné výroby na takzvané chytré továrny a chytrou výrobu. Mnoho současných technologií v Internetu věcí aplikovatelných, kompletní implementace Průmyslu 4.0 však vyžaduje ještě mnoho dalšího výzkumu a vývoje [8].

3.1 Cíle Průmyslu 4.0

Čtvrtá průmyslová revoluce má za úkol implementovat kyberneticko-fyzické adaptivní systémy pro výrobu a pro zavádění nových průmyslových služeb. Do průmyslových produktů a strojů je zaveden inteligentní software, který představuje technologii schopnou předvídat různé události. K tomu se využívá získávání stále nových dat. Tato technologie zajišťuje propojení inteligentních algoritmů s elektronikou a se stroji. To následně zajišťuje strojům schopnosti pro automatickou diagnostiku a pro jejich automatické učení. Inteligentní algoritmy, samostatné učení a inteligence jednotlivých objektů lze také využít pro optimalizaci nabízených služeb. [8]

Důležitá je ovšem myšlenka, že inteligentní chování by se nemělo implementovat jen na samotné stroje, aby nedocházelo k centralizované optimalizaci a inteligenci. Inteligentní chování se musí aplikovat i na okolní objekty, se kterými výrobní stroje přicházejí do styku. Tím se dosáhne inteligentní spolupráce všech objektů. Právě decentralizace inteligence zvyšuje schopnosti samostatného učení a zajišťuje dělání správných rozhodnutí. To vede ke zvýšení celkového výkonu a ke snadnějšímu řízení údržby. Pro transformaci továren na chytré je nutné:

1. Autodiagnostika a samostatné rozhodování strojů: Operátoři výroby, výrobní stroje a samotná výroba jsou dnes řízeny kvalifikovanými inženýry. Vše je směrováno k dodržování termínů, splnění kapacity výroby a k co největšímu vytížení drahých strojů. V této metodě řízení je však opomíjen důležitý fakt a to je údržba. Údržba se ve výrobě prakticky úplně zanedbává a řeší se, až nastane nějaký problém. To následně způsobuje zdržení výroby a je to hlavní příčina nedodržené kapacity výroby a zpoždění termínů. Autodiagnostika a samostatné rozhodování má proto pomoci preventivnímu odstraňování potenciálních problémů ještě před tím, než se začnou vyskytovat a ovlivňovat výrobu.
2. Snížení různorodosti a redundantnosti výrobních strojů: V současných továrnách je velké množství strojů, které jsou si velmi podobné a vykonávají podobnou činnost. Každý stroj je však určen pro odlišný typ výroby a pro odlišný typ výrobku. Unifikace všech zařízení není snadno proveditelná. Implementace

inteligence a samostatného rozhodování do strojů má za úkol propracovat plánování výroby, zjednodušit a unifikovat celou výrobu. Cílem je proto využívání menšího množství strojů, které jsou však schopny adaptace na požadovaný typ výroby bez omezení prostředků.

3. Zvýšení kvality výroby a procesů: Současné stroje nemají možnost informovat vedení výroby neobvyklých událostech, detailech a nesrovnalostech výrobního procesu. Implementace vhodné zpětné vazby od strojů, která by tyto informace poskytovala řídícímu systému pro vyhodnocení a analýzu, má za úkol zlepšit kvalitu výroby a výsledných produktů. V současné době ještě žádná taková technologie neexistuje a je nutné provézt potřebné výzkumy. Cílem zvýšení kvality výroby a procesů je snížení celkového času a nákladů na výrobu požadovaného množství kvalitních produktů.
4. Zpracování velkoobjemových dat a implementace „cloud computing“: Současná infrastruktura internetové komunikace v továrnách neumožňuje zpracování a distribuci velkých množství dat, které jsou nutné pro dosáhnutí samostatného rozhodování, adaptace a autodiagnostiky jednotlivých strojů a zařízení. Pro chytrou továrnu je proto nutné zajistit, s využitím vhodných síťových technologií, robustní síť založenou na „cloud computing“. Zde datový přenos velkých objemů informací nebude způsobovat degradaci výkonu celé továrny a nebude ovlivňovat výrobu negativním způsobem.
5. Zařízení snímací sítě: Senzory představují schopnost strojů vnímat fyzické prostředí a výrobní situace, které je obklopují. Degradace nebo poškození senzorů však může mít fatální důsledky pro kvalitu výroby. V chytré továrně je proto nutné i zajištění algoritmů, které budou ověřovat správnost funkcí všech senzorů.

[8, 9]

3.2 Transformace výroby

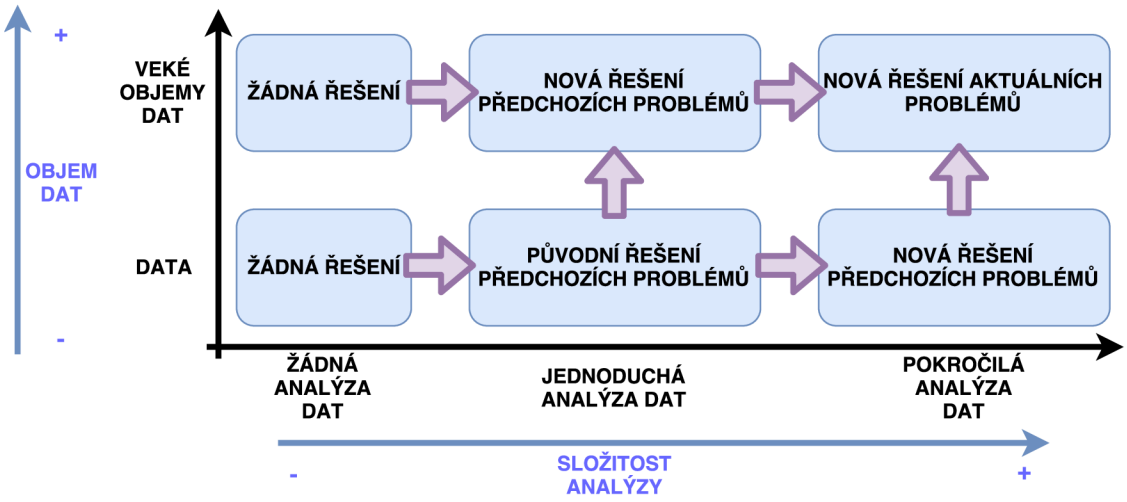
Transformace továren na chytré spočívá ve virtualizaci a individualizované hromadné výrobě. Celá výroba začíná vytvořením digitální kopie požadovaného výrobku, která je asociována se skutečným výrobkem. Tím vznikají kybernetickofyzické systémy, které umožňují hromadnou výrobu individualizovaných výrobků.

V současné době mají pracoviště v továrnách jednoho dirigenta, který zadává jednotlivým strojům úkony, které mají vykonávat. Průmysl 4.0 však implementuje výrobky jako nositele informací pro stroje. Stroje výrobkům pouze poskytují služby. Fyzické výrobky tak budou pomocí svého identifikátoru definovat, jak s nimi má každý stroj, se kterým přijdou do kontaktu, zacházet. V tomto případě budou úkony

definovány ve stroji. Fyzický výrobek však může obsahovat přímo nějaký čip, ve kterém budou všechny informace a úkony pro jednotlivé stroje definovány v digitální podobě. Při kontaktu výrobku se strojem si následně stroj z výrobku načte nastavení a způsob zpracování výrobku.

Průmysl 4.0 implementuje vstup komponent do továrny jako žadatele určitých služeb, které z nich dělají finální výrobky pro export. Výhodou je variabilita výroby, kdy je na každém stroji možno vyrobit prakticky každý požadovaný produkt. Vše samozřejmě v rámci možností a určení každého stroje.

Připojením všech strojů, výrobků a senzorů do Internetu věcí bude pomocí vzájemné komunikace zajištěn trvalý sběr dat. Tato data budou následně sloužit k optimalizaci výroby a výrobních procesů, k usnadnění diagnostiky a údržby a snížení zmetkovitosti. Propojením těchto komponent s informačním systémem se nahradí činnost lidí, a tím se odstraní nedokonalosti lidského faktoru v tomto ohledu „dokonalými“ stroji. Vliv sběru dat a informací na chování strojů znázorňuje obrázek č. 3.1.



Obr. 3.1: Sběr dat je klíčový faktor Průmyslu 4.0 [10].

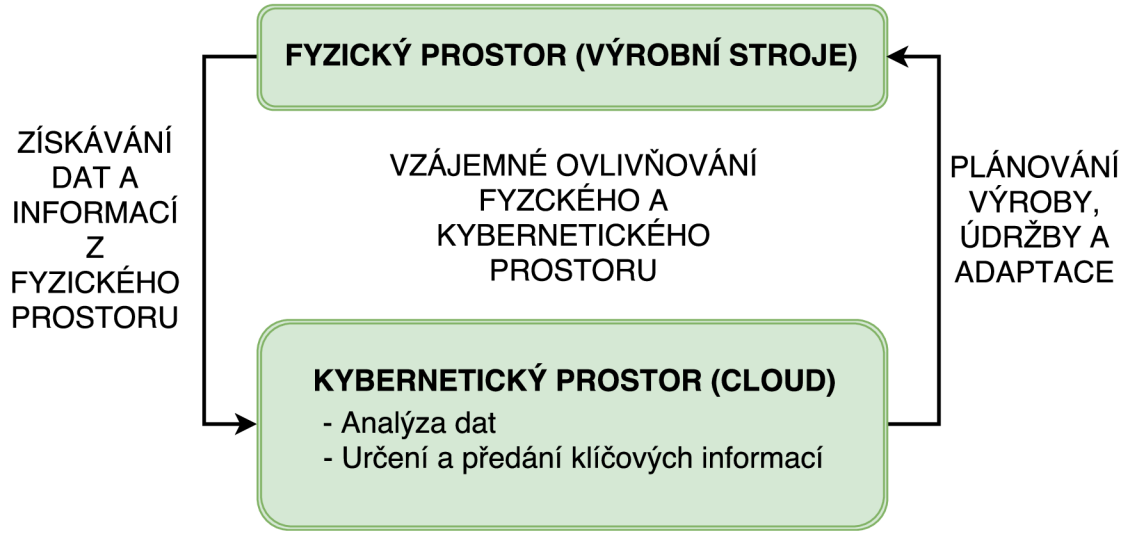
3.3 Přizpůsobení strojů

Jak již bylo zmíněno v předchozím textu, cílem Průmyslu 4.0 je implementace inteligence a schopnosti samostatného rozhodování do výrobních strojů na základě snímání a analýzy dat. Vše za účelem zlepšení plánování výroby, kvality výroby a také automatické diagnostiky strojů. Současné stroje již určitou diagnostikou disponují. Nejsou však schopny problémy předvídat, ale upozorní až v případě, kdy už problémy

nastaly a stroje nejsou dál schopny svých funkcí. Současné stroje také nepředstavují adaptivní a flexibilní zařízení. Transformace stroje do konceptu chytré továrny proto spočívá ve splnění těchto podmínek:

1. Nahrazení současného konceptu strojů, které jsou schopny reagovat jen na operace spuštěné operátorem a nejsou schopny odhalit podmínky, které mají negativní vliv na jejich výkonnost a kvalitu výroby. Nově zavedený koncept chytrých strojů bude umožňovat samostatnou reakci na problémy a nastavování parametrů tak, aby bylo dosaženo univerzálnosti, maximální kapacity a maximální kvality výroby.[8, 9]
2. V současné situaci se stroje učí za pomocí odborníků pomocí dat získaných z experimentů z laboratoří (ladění softwaru, simulace podmínek apod.). Tím se však získá pouze omezený počet dat a není možné odhalit všechny všechny potencionální neobvyklé situace. To se značně rozchází s konceptem Průmyslu 4.0. Implementace schopnosti učit se z analýzy dat získaných v reálném čase při běžném provozu vytváří ze zařízení flexibilní a robustní výrobní stroje s možností adekvátní reakce na různé nepředvídatelné situace. [8, 9]

Pro splnění těchto podmínek by měl být navržen kyberneticko-fyzické rozhraní pro samostatný provoz a samostatnou údržbu, který dokáže efektivně vybrat důležité informace z velkých objemů dat. Na základě těchto dat systém dokáže aplikovat svoji inteligenci do výrobních rozhodnutí. Kyberneticko-fyzické rozhraní je znázorněno na obrázku č. 3.2:



Obr. 3.2: Znázornění kyberneticko-fyzické interakce [8].

Obrázek č. 3.2 znázorňuje „cloud computing“ v Průmyslu 4.0; vztah kybernetické a fyzické části stroje. Kybernetický prostor má za úkol správně definovat data

a informace, která budou následně nasbírána ve fyzickém prostoru. Kybernetický prostor má také za úkol shromažďovat a analyzovat znalosti pro detekci správné funkčnosti zařízení. Získané znalosti sdílí následně s fyzickým prostorem a tím poskytuje fyzickému prostoru stroje data pro vykonávání správných funkcí a rozhodnutí.

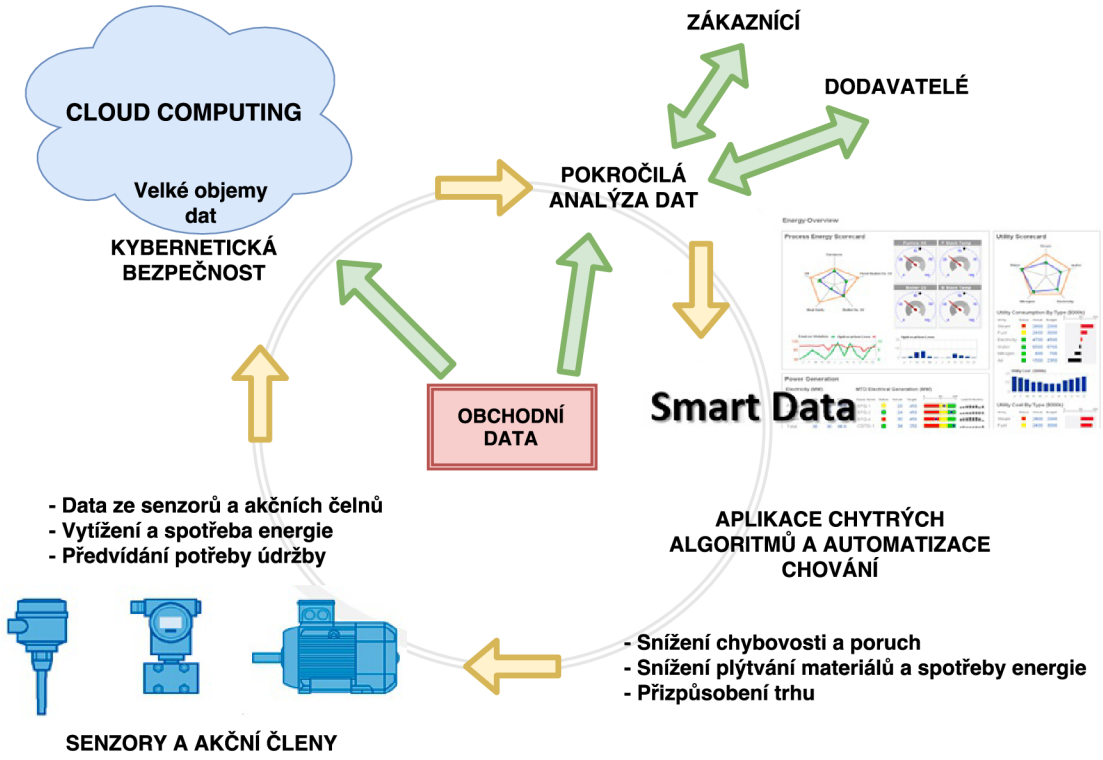
Za fyzický prostor stroje jsou považovány výrobní stroje(sběr dat, výběr parametrů, konfigurace, měření kvality, historie vykonaných úkonů,...) a zásah obsluhy (operace údržby a dohled nad parametry a chodem výroby) [8].

3.4 Shrnutí principů chytré továrny

V chytré továrně je implementovaná plně automatizovaná výroba. Prakticky všechny součásti výroby (skladové zásoby, připravené součásti, finální výrobky, senzory sledující výrobku a výrobní i testovací stroje) jsou vzájemně propojeny pomocí Internetu věcí. Řízení je decentralizované a stroje vykonávají své činnosti na základě vzájemné komunikace. Všechny komunikující prvky shromažďují data, které později využívají ke zkvalitnění a optimalizaci výroby. Funkčnost ideální chytré továrny je pak založena na těchto principech:

- Součásti a komponenty potřebné pro výrobu jsou do továrny transportovány ve chvíli, kdy jsou skutečně potřeba ve výrobě.
- Senzory a případně drony neustále kontrolují skladové zásoby.
- Stroje a roboty vykonávající výrobu spolu nepřetržitě komunikují a vzájemně se informují o nestandardních situacích.
- Součásti a komponenty určené ke zpracování v sobě obsahují digitální informace o tom, jakým způsobem je nutné je zpracovat. Komunikují proto přímo s roboty a výrobními stroji.
- Chytré senzory a testery dohlížejí na kvalitu finálních výrobků.
- Autonomní vozidla poháněná elektrickou energií zajišťují přepravu zásob a finálních produktů po továrně.
- Všechny prvky v chytré továrně pro svou činnost využívají zdrojů alternativní energie.
- Továrna je připojena k Internetu. Disponuje pokročilým zabezpečením proti kybernetickým útokům, které by výrobu ohrozily.
- Zákazník má díky připojení továrny k internetu a díky realizaci uživatelského rozhraní možnost zasahovat do výroby v reálném čase a upravovat ji podle svých potřeb a požadavků.

Principy konceptu Průmyslu 4.0, potažmo chytré továrny, jsou znázorněny na obrázku č. 3.3:



Obr. 3.3: Znázornění principů chytré továrny podle konceptu Průmysl 4.0 [10].

3.5 Současné možnosti

V současné době jsou prakticky všechna výrobní zařízení v továrnách navržena tak, aby byla možná úprava softwarové a případně i hardwarové části. Zařízení jsou prakticky ve všech případech řízena programovatelnými logickými kontroléry PLC. Výhodou programovatelných logických kontrolérů je poměrně snadná úprava softwaru (za předpokladu dobrých znalostí principů funkce stroje nebo dispozice příslušné dokumentace). Průmyslové výrobní zařízení proto nabízí široké možnosti úprav a celkové rekonstrukce. Pokud se jedná o zastaralé průmyslové zařízení, nabízí se jako výhodná možnost výroby nového. Těmito úkony se v současné době zabývá mnoho specializovaných společností v oblasti automatizace a průmyslu.

PLC a jeho technologie poskytují široké spektrum možností výměny dat s dalšími aplikacemi, které běží na zobrazovacích panelech propojených s PLC. Zobrazovací panel proto představuje komunikační rozhraní a zároveň představuje počítač s poměrně velkým výpočetním výkonem pro analýzu dat. Vytvoření sítě, kde je několik výrobních zařízení spojeno s počítačem, na kterém běží aplikace poskytující služby, proto představuje základ pro vytvoření zjednodušené verze Internetu věcí. Toho lze využít pro implementaci konceptu Průmyslu 4.0. K tomu je však nutné

navrhnout a realizovat komplexní protokol, který bude řešit komunikaci, sběr dat a poskytování služeb ve vytvořené síti.

Průzkumu možností komunikace a sběru dat ze zařízení řízených PLC, úpravě softwaru stávajících zařízení s případnou úpravou hardwaru, navržení protokolu komunikace a vytvoření aplikace s možnostmi sběru a ukládání dat do úložišť se věnují následující kapitoly této práce.

4 MOŽNOSTI SBĚRU DAT A KOMUNIKAČNÍCH TECHNOLOGIÍ VYUŽITELNÝCH V PRŮMYSLU 4.0

V této kapitole je stručně představen programovatelný logický kontrolér a jeho komunikační možnosti. Následně jsou zde představeny technologie využitelné pro sběr informací ze senzorů programovatelného logického kontroléru a technologie pro připojení programovatelného logického kontroléru do sítě chytré továrny. V poslední části je představena možnost ukládání dat získaných ze sítě chytré továrny.

4.1 Programovatelný logický kontrolér

Programovatelný logický automat, ve zkratce PLC, se v průmyslové automatizaci a v oboru měření a regulace využívá pro řízení aplikací a procesů. Z hlediska elektroniky je PLC navržen tak, aby bylo možné přímé připojení akčních členů a senzorů bez nutnosti doplnění dalších elektronických prvků. Je možné využívat jak digitální, tak analogové vstupy/výstupy. PLC tedy řídí celý proces prakticky libovolného zařízení: řídí činnost akčních členů a zaznamenává data z příslušných senzorů.

PLC představuje procesor, který v prvním kroku načte hodnoty všech svých vstupů. V druhém kroku čte z paměti a následně vykonává programátorem definovanou posloupnost instrukcí. Instrukce tedy reprezentují implementovaný software. V posledním kroku provádí zápis na všechny svoje výstupy. Tyto kroky neustále cyklují. Vše je znázorněno na obrázku č. 4.1.

Kromě analogových a digitálních vstupů/výstupů, PLC také disponuje Ethernetovým rozhraním, případně dalšími komunikačními rozhraními (RS-232, RS-485,...). Díky Ethernetovému rozhraní je možné PLC připojit k Internetu. Tím se také otevírá široká škála možností, jakým způsobem potřebná data z PLC získat a následně zobrazit v aplikaci na příslušném zobrazovacím panelu, který je také do Internetu připojen. Existuje také možnost potřebná data z aplikace do PLC odeslat. K tomuto účelu slouží OLE[1] for Process Control neboli OPC server, který zajišťuje usnadnění spolupráce jednotlivých softwarových aplikací v průmyslu a v oblasti řízení procesů. Technologii OPC se dále věnuje následující podkapitola.

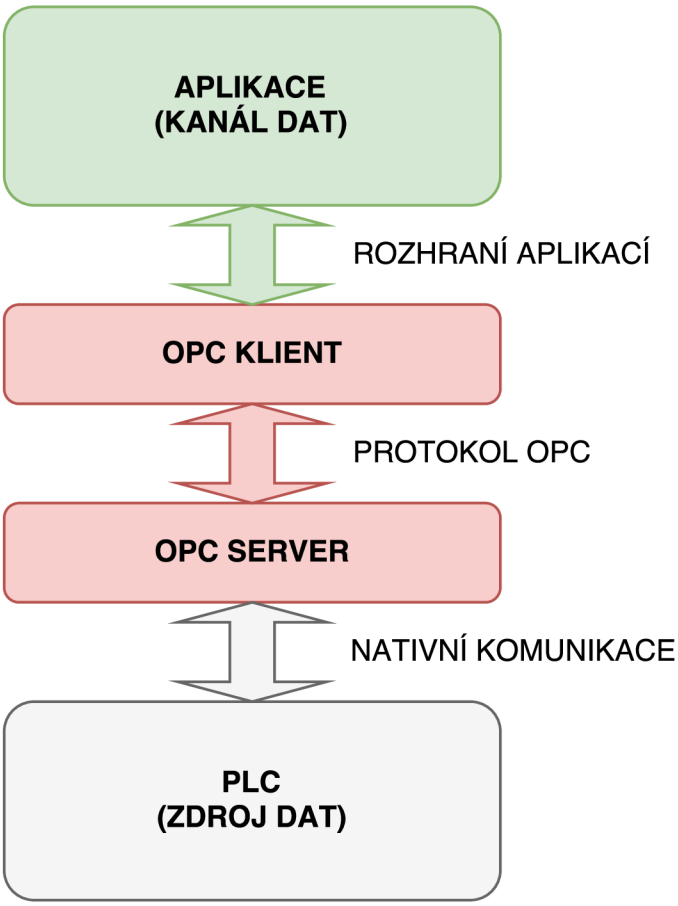
[1]OLE = Object Linking and Emnedding, jedná se o technologii pro vkládání, propojování a úpravu objektů.

Obr. 4.1: Princip činnosti programovatelného logického automatu.

4.2 Technologie OPC

Při vytváření průmyslových aplikací, jejichž cílem je komunikovat s PLC (tj. zobrazovat a nastavovat data), je nutné naprogramovat komunikační vlákno mezi softwarem (aplikací) a hardwarem (PLC). K tomuto účelu je za normálních podmínek nutné využívat ovladače, které jsou pro každý hardware odlišné. Často je nutné příslušný ovladač naprogramovat. S využitím OPC však tyto starosti odpadají a odpadá i nutnost znalosti nativních komunikačních protokolů a vnitřní organizace dat. OPC představuje komunikační technologii, jejímž cílem je vytvoření snadného a jednotného komunikačního rozhraní mezi hardwarem a softwarem používaným v průmyslové automatizaci. OPC reprezentuje abstraktní vrstvu mezi zdrojem a příjemcem dat, viz obrázek č. 4.2. [14, 15]

Komunikace s využitím OPC je založena na principu klient-server, kde klient i server jsou aplikovány softwarově. OPC klient přijímá data z OPC serveru v OPC protokolu a prezentuje je pro cílovou aplikaci. OPC server komunikuje s připojenými PLC pomocí nativních komunikačních protokolů a transformuje komunikaci do OPC protokolu.



Obr. 4.2: Princip výměny dat technologie OPC [14]

4.2.1 Typy OPC

Data určené k přenosu mezi PLC a aplikací lze rozdělit do 3 kategorií. Jedná se o data reálného času, dřívější data a data alarmů a událostí. Každý typ dat je nositelem určitého rozsahu informací a datových typů. OPC umožňuje přenášet všechny tři kategorie bez nutnosti definice, jakým způsobem je který typ přenášen. Pro pokrytí přenosu všech kategorií OPC může tedy pracovat v těchto 3 režimech:

1. OPC Data Access (OPC DA) pro přenos dat v reálném čase.
2. OPC Historical Data Access (OPC HDA) pro přenos dat časově nekritických.
3. OPC Alarms & Events (OPC AE) pro přenos informací obsahujících upozornění, varování a chybová hlášení.

[14]

4.2.2 Výhody OPC

1. Aplikace založené na OPC můžou prakticky bez omezení komunikovat s každým zařízením libovolného výrobce, které také podporuje OPC komunikaci. Odpadá nutnost implementace ovladačů různých pro každého výrobce zařízení.
2. I když je často v OPC severu omezen počet možných připojených zařízení, tak jediný OPC systém umožňuje propojení více aplikací s více zařízeními.
3. Většina současných průmyslových zařízení a většina používaných programovacích jazyků má stále rostoucí podporu aplikací založených na technologii OPC. Využití OPC je proto velmi snadné a elegantní řešení.

[14]

4.3 Nativní komunikační protokoly

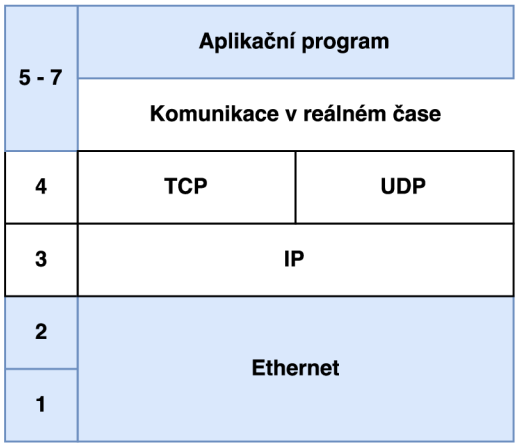
Každý výrobce PLC má svůj systém založený na proprietárním komunikačním protokolu. Tento protokol následně využívá pro komunikaci s různými ostrovními systémy (vzdálené I/O, ventilový blok, ...), které využívá pro řízení procesů. Kromě komunikace s dalšími zařízeními se proprietární protokol využívá nejčastěji také k výměně dat a informací mezi PLC a OPC serverem výrobce. V uživatelské aplikaci tak odpadá nutnost znalosti proprietárních protokolů a není nutné tuto komunikaci doprogramovat. Právě na využití nativních komunikačních protokolech s vhodným rozhraním bude založen návrh a implementace chytré továrny podle konceptu Průmysl 4.0. Z tohoto důvodu jsou v této podkapitole představeny proprietární komunikační protokoly nejrozšířenějších PLC různých výrobců, kterých se potencionálně implementace chytré továrny bude týkat.

Mezi velké výhody těchto komunikačních protokolů patří, že pro komunikaci velkého množství zařízení stačí pouze jedna datová sběrnice. Některé nativní komunikační protokoly jsou založeny na standardu Ethernet, proto je zde také zkráceně popsán pro ucelení představy o ostatních komunikačních protokolech. [8]

4.3.1 Ethernet

Ethernet je označení technologie, která se v dnešní dobře používá v běžných počítačových sítích. Ethernet je standardizovaný jako IEEE 802.3. O Ethernetu se hovoří jako síti pro použití v kancelářském prostředí a v prostředí internetu (Ethernet TCP/IP). Jeho vývoj do současné podoby ho umožňuje využívat v různých modifikacích také v řídicích systémech v průmyslu a automatizaci. V Ethernetu TCP/IP se časově kritická (data reálného času) data i časově nekritické zprávy přenášejí

protokoly TCP/UDP/IP. Komunikační model Ethernetu je znázorněn na obrázku č. 4.3. [8, 9]



Obr. 4.3: Komunikační model Ethernet [9].

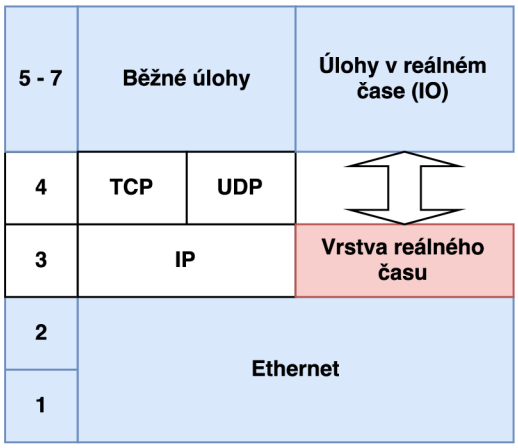
Klasický komunikační protokol Ethernet TCP/IP je využíván pro komunikaci mezi MELSEC OPC a PLC společnosti Mitsubishi Automation.

4.3.2 Profinet

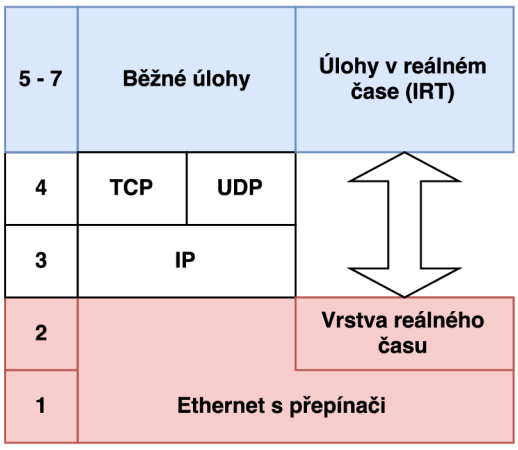
Profinet byl vyvinut organizaci PNO (Profibus Nutzerorganisation) ve spolupráci se společností Siemens. Existuje ve dvou verzích. U Profinetu verze 2, neboli Profinetu IO probíhá komunikace ve dvou kanálech. V prvním kanálu jsou standardní zprávy bez požadavků na přenos v reálném čase přenášeny standardní cestou s využitím TCP/UDP/IP. Druhý, paralelní kanál obsahuje softwarové překlenutí vrstev 3 a 4 vrstvového modelu, takže lze dosáhnout dokonalejších vlastností reálného času. K dalšímu zlepšení vlastností reálného času je u Profinetu redukována délka přenášeného bloku dat a je zaveden mechanismus určení priority podle standardu IEEE 802.1p (až do priority 7 u komunikace v reálném čase). [9]

U Profinetu verze 3, neboli Profinetu IRT, je pro vrstvy Ethernetu použit speciální hardware realizující hardwarové překlenutí vrstev TCP/IP. Profinet IRT je určen pro úlohy probíhající v reálném čase s přísnými požadavky na dodržení doby odezvy a synchronizace. Profinet ve verzi 3 může pracovat v klasické přepínané síti Ethernet. Je vhodný k řízení např. ventilových bloků či pneumatických pohonů v průmyslové automatizaci. Přenos běžných zpráv bez požadavků na přenos v reálném čase, včetně přístupu k internetu, je zajištěn paralelní cestou TCP/UDP/IP. [9]

Na Profinetu je založen protokol S7, který zajišťuje výměnu dat mezi PLC S7-1200 nebo S7-1500 a OPC serverem Simatic NET od společnosti Siemens.



Obr. 4.4: Komunikační model Profinet IO [9].



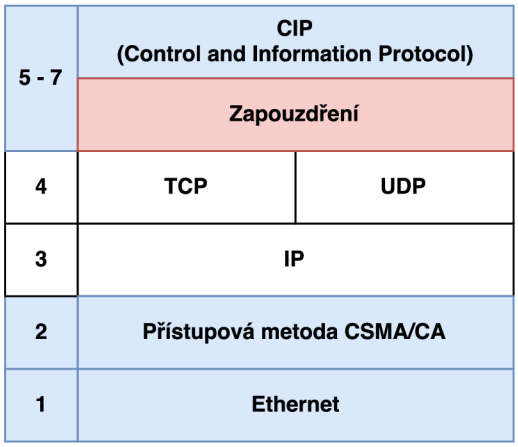
Obr. 4.5: Komunikační model Profinet IRT [9].

4.3.3 Ethernet/IP

Sběrnice Ethernet/IP je založena na Ethernet TCP/IP. Pro přenos dat může využívat jak protokolu TCP, tak i protokolu UDP. Zkratka IP v názvu indikuje průmyslový protokol (Industrial Protocol) založeného na standardu Ethernet. Jeho model představuje rozšíření komunikačního modelu pro průmyslovou komunikaci. Ve vrstvě 7 modelu se nachází protokol aplikační vrstvy CIP[2], vytvořený pro průmyslové sítě DeviceNet a ControlNet a v systému ETHERNET/IP použitý nad protokoly TCP/IP. Necyklická data (konfigurační a parametrizační data, nahrání programu apod.) jsou zabezpečeným způsobem přenášena spojovanou službou protokolem

[2]CIP = Control and Information Protocol, jedná se o protokol zajišťující komunikaci a služby pro aplikace určené do průmyslové automatizace.

TCP, zatímco časově kritická data jsou přenášena nespojovanou službou protokolem UDP. [9]



Obr. 4.6: Komunikační model Ethernet/IP [9].

Na protokolu Ethernet/IP je založena komunikace mezi PLC Allen-Bradley a OPC serverem RSLinx od společnosti Rockwell Automation.

4.3.4 EPSNET

Síť EPSNET může být založena na komunikačním protokolu RS-485 nebo také na Ethernetové síti. Jedná se o proprietární protokol společnosti Teco a.s. Komunikace je vyvolávána nadřízenou stanicí na principu dotaz - odpověď. Tento princip umožňuje připojení většího počtu účastníků k nadřízené stanici na poloduplexním rozhraní. Na síti EPSNET mohou pracovat dva druhy stanic:

- Nadřízená stanice (master) - aktivní účastník, řídí komunikaci,
- podřízená stanice (slave) - pasivní účastník, odpovídá na dotazy nadřízené stanice.

[13]

Ethernetové verze protokolu EPSNET se označuje EPSNET UDP je založena na protokolu UDP. Komunikace založená na síti EPSNET může být rozdělena až do 126 kanálů. To znamená, že je možné komunikovat až se 126 zařízeními. [13]

Model komunikace je totožný s modelem komunkace klasického Ethernetu uvedeného v časti 4.3.1. Na protokolu EPSNET je založena komunikace PLC s OPC serverem PLCComS od společnosti Teco a.s.

4.4 Použití databázových systémů v průmyslové automatizaci

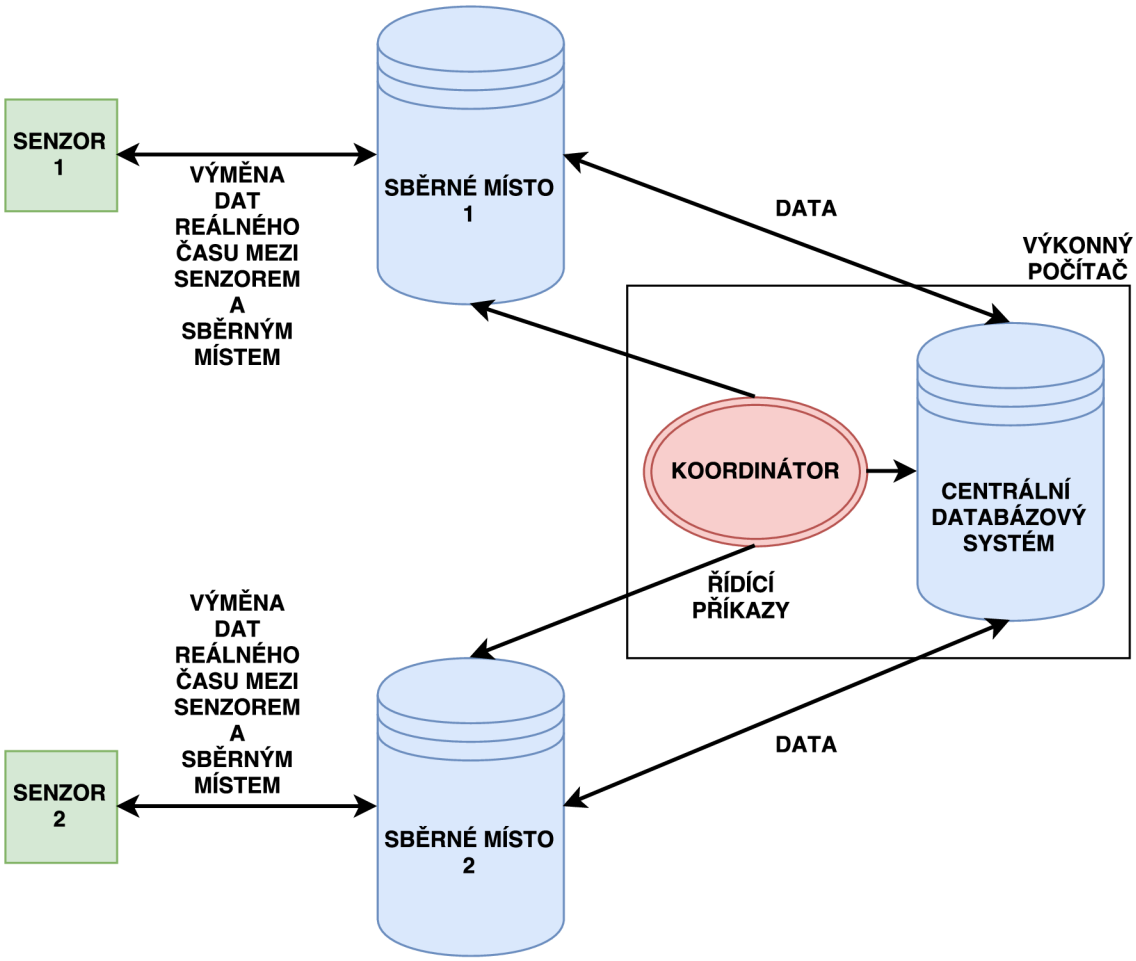
S implementací Průmyslu 4.0 do továren, který je mimo jiné založen právě na neustálém sběru velkého množství dat, vzniká požadavek na schopnost systému spolehlivě a efektivně uchovávat velké množství dat. Technologii, která dokáže tento požadavek splnit, představují databázové systémy. Databázový systém sám o sobě umožňuje uchovávání velkého množství dat nezávisle na použitých technologiích. Nabízí velké množství funkcí a vlastností, které lze bez pochyb využít i v průmyslové automatizaci:

1. Nízké nároky na výpočetní výkon: Všechny operace s daty provádí databázová logika, které je navržena tak, aby zabírala co nejméně výpočetních prostředků. Aplikace následně k práci s daty využívá databázové logiky, která jí poskytuje pouze výsledky operací.
2. Nezávislost na použitých technologiích: Data jsou ukládána do jednoduchých tabulek. Do těchto tabulek tak může pomocí databázové logiky přistupovat jakákoliv aplikace, která potřebuje s daty pracovat.
3. Minimální nadbytečnost dat: Protože jakákoli aplikace může do databázové tabulky přistupovat a získávat potřebná data, není nutné data duplikovat pro každou aplikaci samostatných souborů.
4. Další výhody databázových systémů: Sjednocení dat, lepší kontrola a údržba dat, snadnější sdílení dat, jednoduchý dohled na data, snadné převedení dat na potřebný formát a zjednodušení vývoje aplikací.

[16, 17]

Klasické databázové systémy jsou určeny pro ukládání trvalých dat. V průmyslové automatizaci, tím i v chytré továrně, je však nutné zaznamenávat data s omezenou dobou trvání než dojde k jejich změně (hodnoty senzorů, zaznamenané výsledky apod.). Tato data se označují jako data reálného času. Kvůli datům reálného času vznikají i databázové systémy reálného času. Mezi jejich nejdůležitější kritéria patří schopnost zpracovat data u určitě definovaném čase. Pokud by se data nezpracovala v daném intervalu, může to kritickým způsobem ovlivňovat celý proces. Databázový systém proto musí mít implementované tzv. řízení kvality služeb, které musí řídit prioritní systém zaznamenávání dat tak, aby data časově kritická byla zpracovávána přednostně. [16]

Jako vhodné řešení zaznamenávání dat v reálném čase do databáze se nabízí systém, ve kterém existuje několik tzv. sběrných míst. Tato sběrná místa mají neustálý přístup ke kanálům, do kterých ostatní zařízení (například senzory) neustále svá data odesílají. Jednotlivá sběrná místa mají dále pevně definováno, jaký typ dat

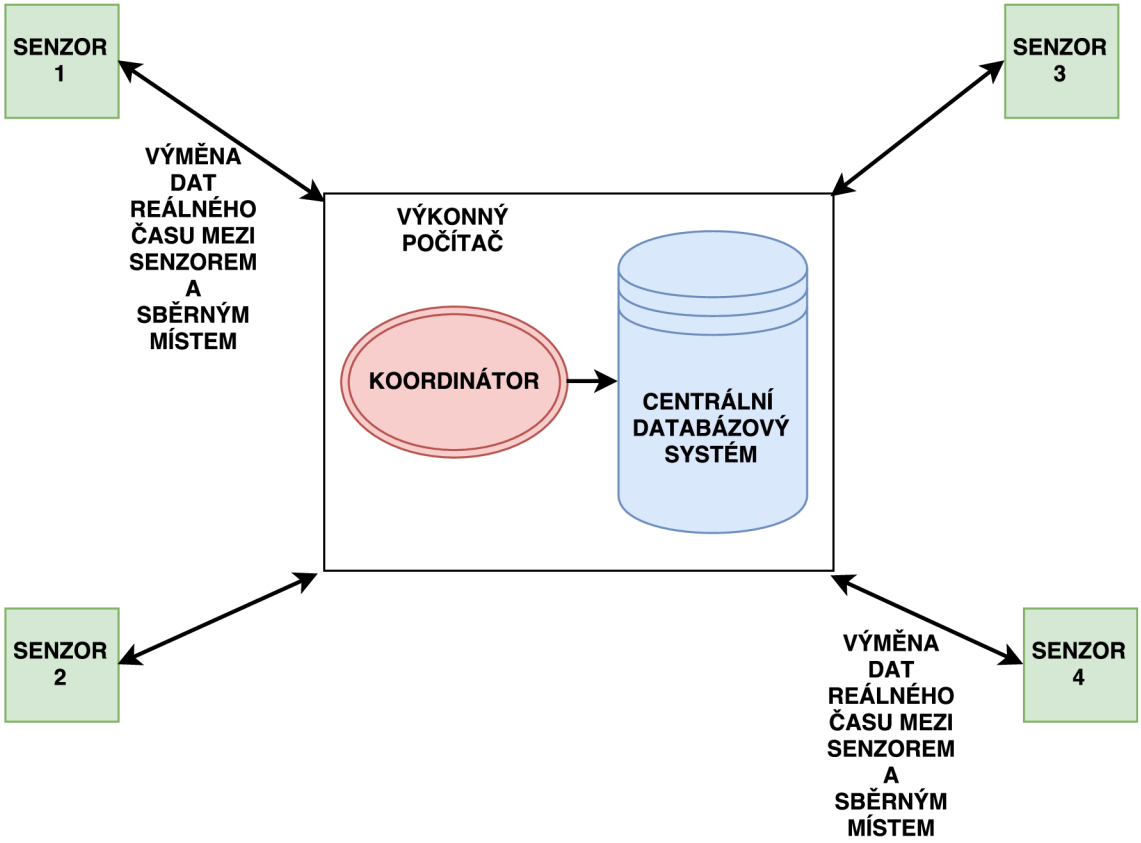


Obr. 4.7: Příklad databázového systému reálného času.

a informací a jakým způsobem mají zaznamenávat (například pevně daný časový okamžik, specifický signál a pod.). Sběrné místo si tedy udržuje svoji databázi reálného času a je schopno jednoduché analýzy dat. Zaznamenávaná data jsou dále poskytována centrální databázi, kde již nejsou časové požadavky na zaznamenávání dat. Koordinátor dále data poskytuje do centrální databáze pro další analýzu a vyhodnocení dat. Na komunikaci mezi sběrnými místy a centrálním systémem dohlíží tzv. koordinátor. Princip takového databázového systému reálného času je schématicky zobrazen na obrázku č. 4.3. Na podobném principu je založen open-source systém Druid od společnosti Matemarkets. [17]

Databázový systém reálného času lze však realizovat i za pomocí jednoho sběrného systému, ke kterému jsou připojeny všechny periferie sbírající data a také periferie provádějící akční zásahy. V tomto databázovém modelu hraje velkou roli přesná definice a optimalizace výměny informací. Optimalizace spočívá ve filtrování dat na data, která je nutná zpracovávat v reálném čase, a dále na data, jejichž zpraco-

vání není časově kritické a má menší prioritu. Příklad tohoto systému je znázorněn na obrázku č. 4.8.



Obr. 4.8: Další říklad databázového systému reálného času.

Databázový systém již bývá součástí různých komunikačních technologií pro komunikaci mezi PLC a uživatelskou aplikací. OPC tak již může být doplněno systémem databázového systému reálného času. Výrobce v těchto případech garantuje kvalitu poskytovaných funkcí a technologií, které zajišťují sběr velkoobjemových dat v reálném čase, přístup do databáze v reálném čase a rychlé vyhledávání. Mezi zástupce těchto technologií patří WINCC a WINAC od společnosti Siemens, který obsahuje databázový systém Sybase SQL AnyWhere nebo RSViewStudio s SQL AnyWhere nebo Dbase od společnosti Rockwell Automation (Allen-Bradley PLC).

5 STÁVAJÍCÍ SITUACE V TOVÁRNÁCH

Před implementací systému pro Průmysl 4.0 je důležité seznámit se se současnou situací v továrnách. Právě proto se tato kapitola zaměřuje na seznámení se stávajícím stavem ve vzorové továrně. Vzorová továrna vychází ze skutečné situace a lze ji aplikovat prakticky na každou továrnu, která se zabývá průmyslovou výrobou nejrůznějších komponent.

Nejprve jsou obecně popsány výrobní postupy v továrně a postupy s výrobou související. Následuje popis testovacích zařízení, která jsou prakticky posledním článkem výroby, a která provádí výstupní kontrolu výrobků. Tato testovací zařízení budou později jedním z hlavních cílů implementace automatizovaného chování určeného pro uplatnění v chytré továrně. V poslední části této kapitoly je představena motivace pro implementaci systému chytré továrny.

5.1 Popis výroby

Cílem vzorové továrny je výroba dílů pro automobilový průmysl a výroba součástí s elektrickým pohonem. V továrně se na jednotlivých pracovištích provádí kompletace finálních zařízení z komponent vyráběných v jiné divizi společnosti, případně jsou komponenty získávány od různých dodavatelů. Z tohoto důvodu se tato podkapitola zaměřuje na kompletaci finálních výrobků. Celý proces kompletace finálního produktu se skládá z těchto úkonů:

1. Příprava jednotlivých částí,
2. nanesení emulzí na příslušné části,
3. sestavení jednotlivých komponent,
4. svaření spojů sestaveného výrobku v svářečce,
5. otestování svařeného výrobku v testovacím stroji (testeru),
6. zabalení produktu a příprava na export produktu k zákazníkovi.

Výše uvedený postup je téměř totožný u všech továrnou vyráběných komponent. Výrobní postup lze konkretizovat na výrobě palivového filtru, který je v této práci uveden jako příklad:

1. Příprava jednotlivých částí (filtrová vložka, plastový/hliníkový obal filtrové vložky a víko obalu),
2. promazání vývodů filtrové vložky příslušnou emulzí,
3. vložení filtrové vložky do obalu a zakrytí víkem,
4. svaření spojů složeného produktu v příslušné svářečce (například s využitím ultrazvukového svařování),
5. osazení vývodů filtru záslepkami,

6. otestování těsnosti svařeného výrobku v testeru,
7. zabalení produktu do krabice a příprava na export produktu k zákazníkovi.

Celý proces kompletace produktu v současné době realizují manuálně operátoři na jednotlivých stanovištích, případně jsou některé části procesu kompletace převedeny do určité míry na automatizované: Například operátor vloží sestavený filtr na dopravník, který ho odveze k automatické svářečce. Následně je produkt založen robotem do svářečky a svařen. Za jednu z nejdůležitějších částí procesu kompletace lze považovat výstupní testování výrobku. Princip testování je představen v následující podkapitole.

5.2 Testování produktu

V továrně jsou vyráběny produkty, které jsou určeny jednak k průtoku nějaké kapaliny popřípadě plynu nebo jsou určeny ke stálému naplnění kapalinou případně plynem (například průtok nafty přes palivový filtr nebo naplnění schránky s elektrickým pohonem uvnitř plynem). Z tohoto důvodu musí výrobce garantovat definovanou těsnost vyrobeného produktu tak, aby nedocházelo k nežádoucím ztrátám této kapaliny nebo plynu. K tomuto účelu obsahuje modelová továrna i několik stanovišť (samostatných stanic neboli testerů), kde je právě těsnost vyrobených produktů ověřována příslušnými měřicími metodami.

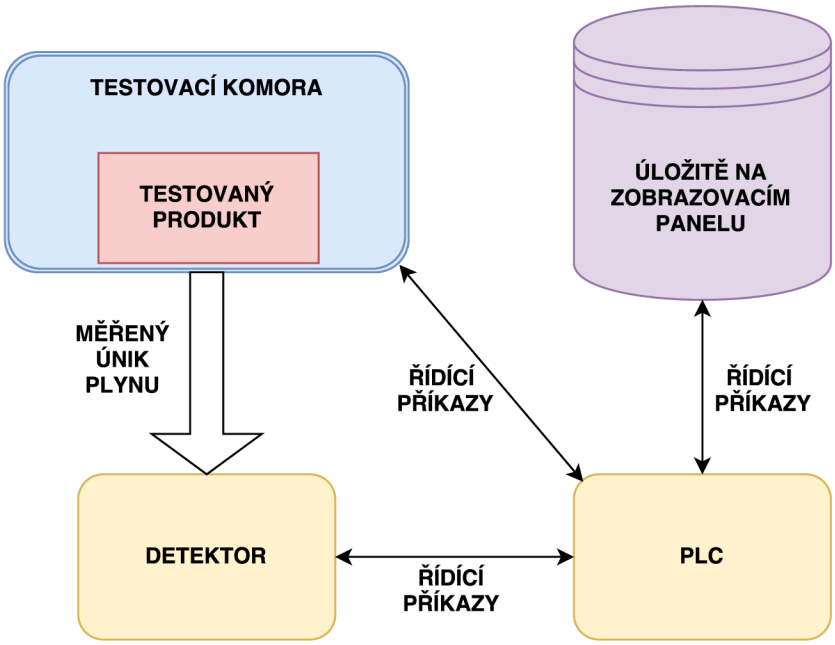
5.2.1 Obecný popis měřicích stanic

Každá měřicí stanice představuje poloautomat umožňující manuální spouštění několika automatických funkcí. Je řízený programovatelným logickým kontrolérem PLC. PLC může potom být, v závislosti na dodavateli stanice, vyrobeno od různých výrobců (nejčastěji však Siemens, Teco, Mitsubishi nebo Allen-Bradley). Měřicí stanice pak může být v závislosti na požadovaném taktu testování jednokomorová případně vícekomorová. Pro vysvětlení: Na vícekomorové stanici je možné spustit několik testů těsnosti současně, vždy podle počtu komor, ve kterých je možno vykonávat měření. Například na 3 komorové stanici lze současně testovat až 3 výrobky. Tím dojde ke značnému zvýšení taktu testování.

Stanice pro měření těsnosti může být založena například na principu měření poklesu/nárůstu tlaku. V tomto případě se měří změna tlaku v produktu, který je natlakovaný vzduchem. Další (přesnější) metoda testování těsnosti je založena na principu naplnění produktu héliem a měřením úniku hélia hmotnostním spektrometrem. Při tomto typu testování se využívá skutečnosti, že hélium je inertní plyn s velmi malými molekulami, které projdou i velmi malými netěsnostmi. V obou případech však měřicí stanice pracuje velmi podobným způsobem.

Stanice pro testování těsností výrobku je konstruována tak, aby byla schopna testovat několik typů vyráběných produktů. Podle požadavku na měření výrobku, je nutné v testovací stanici zvolit parametry měření a tím měřený produkt nějakým způsobem definovat tak, aby stanice věděla jakým způsobem má měřit a dle jakých parametrů testy následně vyhodnocovat. Operátor případně seřizovač stanice musí vždy podle produktu adekvátně zvolit měřicí metodu na operátorském panelu.

Testovací stanice je zjednodušeně znázorněna na obrázku č. 5.1:



Obr. 5.1: Testovací stanice.

5.2.2 Postup testování výrobku ve stanici

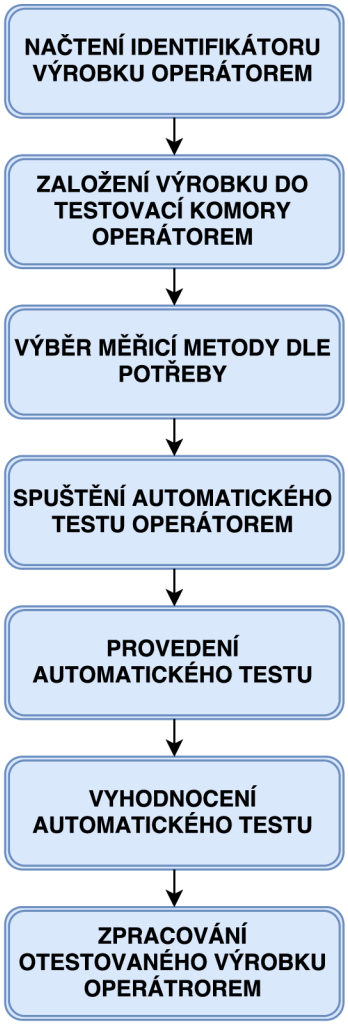
Definice měřeného výrobku se zpravidla provádí instalací vhodných adaptérů pro připojení výrobku k měřícímu systému a ve výběru měřicí metody, která kromě dalších informací definujících test obsahuje i limitní hodnotu netěsnosti pro měřený produkt. Pokud jsou adaptéry pro připojení měřeného výrobku konstruovány jako nevyměnitelné a univerzální, může být výběrem měřicí metody definován i způsob připojení měřeného výrobku.

Po zvolení měřící metody operátor vždy načítá identifikátor výrobku (tzn. identifikační kód, pod kterým je výrobek uložen s výsledkem testu do databáze výsledků měření). Po načtení identifikátoru je výrobek založen do prostoru měření (tzv. měřicí komory) a následně manuálně nebo po spuštění testu automaticky připojen do měřicího systému. Stiskem startovacího tlačítka se spustí automatický cyklus měření

netěsnosti. Po spuštění testu se měřicí prostor (komora) uzavře a vyevakuuje vakuovou vývěvou tak, aby byla zajištěna definovaná hodnota pozadí v měřicí komoře. Výrobek se následně naplní požadovaným plynem (vzduchem nebo héliem) na požadovaný tlak. Po stanovený čas se následně měří únik plynu z výrobku do uzavřené komory, která je spojena s příslušným detektorem. Množství uniklého plynu z výrobku do měřicí komory pak představuje výslednou netěsnost testované komponenty.

Po skončení měřicího cyklu je v PLC testeru určen výsledek (OK = těsný, NOK = netěsný) a tester ukládá výsledek do své databáze. Operátor následně otestovaný kus posílá k dalšímu stanovišti zpracování (výsledek OK) případně ho vyhazuje do zmetkovníku (výsledek NOK).

Celý proces je ještě jednou shrnut v blokovém diagramu, viz obrázek č. 5.2.



Obr. 5.2: Postup testování výrobku.

5.3 Motivace pro přizpůsobení strojů konceptu chytré továrny

V modelové továrně se nachází celkem 4 zařízení, které provádí héliový test netěsnosti. Není však žádným způsobem zajištěno porovnávání a analýza dat z jednotlivých strojů. Z tohoto důvodu vznikl požadavek na zajištění následujících 3 úkonů:

1. Protože jednotlivá testovací zařízení testují podobné nebo v mnoha případech i stejné produkty, potřebuje mít vedení výroby (případně vedení společnosti a koncový zákazník) ucelené informace o výsledcích otestovaných produktech stejného nebo podobného typu ze všech testovacích zařízení. Z tohoto důvodu vznikl požadavek na centralizovaný sběr výsledků, ze kterých bude umožněno získávat kompletní informace o výsledcích všech testovaných produktů nezávisle na použitém stroji.
2. Další motivací je zlepšení v oblasti údržby. S tím souvisí jednak monitorování a vyhodnocování opotřebení jednotlivých strojů (sledování moto-hodin nebo sledování počtu pracovních cyklů). Dále také monitorování a vyhodnocování výsledků a hodnot senzorů za účelem včasné výměny spotřebního materiálu ovlivňující kvalitu testování (například těsnění).
3. Je nutné brát v úvahu i fakt, že každý testovaný produkt může mít odlišné požadavky na nastavení parametrů prováděného testu. Obsluha stroje nemusí vždy vybrat správnou měřicí metodu. Kvůli tomu vzniká požadavek na to, aby si stroj podle produktu sám určil měřicí metodu a tím se výběr zrychlil a také se odstranila chybovost lidského faktoru při výběru. Stroj by si měl proto podle načteného kódu produktu automaticky vybírat měřicí metody. Motivací je proto i zajištění správnosti nastavení měřicího stroje.

6 NÁVRH STRUKTURY SYSTÉMU PRO KONCEPT CHYTRÉ TOVÁRNY

V této kapitole je popsán návrh struktury systému, který je určen pro implementaci do chytré továrny. V jednotlivých částech jsou popsány jednotlivé cíle, které musí navržená struktura splňovat, dále je uveden popis potřebných úprav v měřicích zařízeních, je popsán návrh komunikační sítě spojující jednotlivé prvky, je také popsán koncept cílové aplikace a nakonec je popsán návrh celé komunikace s ohledem na bezpečný a plynulých chod všech zařízení. Výběr technologií a jednotlivé úkony návrhu struktury systému jsou provedeny s ohledem na zadání práce, s ohledem na charakter zařízení a s ohledem na požadavky třetích stran, kterých se implementace týká.

6.1 Cíle implementace

Navrhovaný systém pro chytrou továrnu zahrnuje úpravu stávajících měřicích zařízení, připojení stávajících zařízení do společné komunikační sítě a vytvoření aplikace, která bude jednotlivým zařízením a osobám v továrně zajišťovat požadované služby. Všechny cíle implementace jsou uvedeny v následujících bodech:

- Doplnění hardwarových komponent do stávajících měřicích zařízení.
- Doplnění softwaru nebo kompletní předělání softwaru ve stávajících měřicích zařízeních.
- Návrh a realizace aplikace, která bude zajišťovat plošný sběr dat ze všech zařízení a bude poskytovat požadované služby všem zařízením a osobám v továrně.
- Propojení jednotlivých zařízení tak, aby byla schopna poskytovat a přijímat data podle potřeb.
- Návrh a realizace komunikačního protokolu, podle kterého budou zařízení v síti s aplikací komunikovat.

6.2 Úprava stávajících měřicích zařízení

V továrně jsou celkem 4 zařízení, která zajišťují měření těsnosti vyráběných produktů s využitím héliového testu. Na měřicích zařízeních bude nutné doplnění některých hardwarových komponent. Také bude nezbytný zásah do softwaru. Většina těchto úkonů úpravy softwaru v PLC systémech je mimo rozsah této práce, proto se tato podkapitola bude zabývat pouze návrhem doplnění hardwarových a softwarových prvků, které s touto prací souvisí.

Zde je v několika bodech uveden krátký přehled těchto zařízení s úkony, které je nutné do měřicích zařízení doplnit. Pro jejich implementaci do chytré továrny je nutné zajistit:

1. **Měřicí stanice S7-1215C a S7-1214C**: Jedná se o 2 dvoukomorové měřicí systémy řízené PLC Siemens S7-1215C a PLC Siemens S7-1214C. Do obou stanic je nutné doplnit:
   - Doplnění softwaru o počítadlo motohodin.
   - Doplnění softwaru o počítadlo měřicích cyklů včetně počítadla OK kusů.
   - Úprava parametrů v měřicí metodě a doplnění rozhodovacího algoritmu pro automatické vybírání a nastavování měřicí metody.
   - Hardwarová a softwarová implementace čtečky 2D kódů testovaných produktů. Pro každou komoru stanice bude zvolena čtečka Keyence SR-1000, se kterou bude PLC komunikovat pomocí proprietárního protokolu Profinet.
2. **Měřicí stanice Mitsubishi Q**: Nejnovější stanice řízená PLC Mitsubishi řady Q26. Jedná se o 2-komorový systém. Stanice již disponuje čtečkou kódů (Keyence SR-1000, komunikace pomocí proprietárního protokolu MELSEC), počítadlem pracovních cyklů jednotlivých komponent, počítadlem testovacích cyklů a počítadlem motohodin. Nutné je doplnění pouze rozhodovacího algoritmu pro automatické vybírání a nastavování měřicí metody.
3. **Měřicí stanice Foxtrot**: Jednokomorová měřicí stanice řízená PLC Teco Foxtrot CP-1005, nutno doplnit:
   - Doplnění softwaru o počítadlo motohodin.
   - Doplnění softwaru o počítadlo měřicích cyklů včetně počítadla OK kusů.
   - Úprava parametrů v měřicí metodě a doplnění rozhodovacího algoritmu pro automatické vybírání a nastavování měřicí metody.
   - Hardwarová a softwarová implementace čtečky 2D kódů testovaných produktů. I vzhledem k použitelnému komunikačnímu protokolu (využití klasického Ethernetu, vytvořeno TCP spojení mezi čtečkou a PLC Foxtrot CP-1005) byla vybrána čtečka Datalogix Matrix 210N.

6.3 Aplikace pro nadřazený systém

Cílová aplikace pro nadřazený systém celého systému obsahuje 3 klíčové části: grafické rozhraní pro zobrazování požadovaných hodnot, prostředky pro komunikaci s měřicími stroji a prostředky pro vytvoření potřebného databázového systému a práci s ním.

Pro vytvoření aplikace pro nadřazený systém byl na základě požadavků na jed-

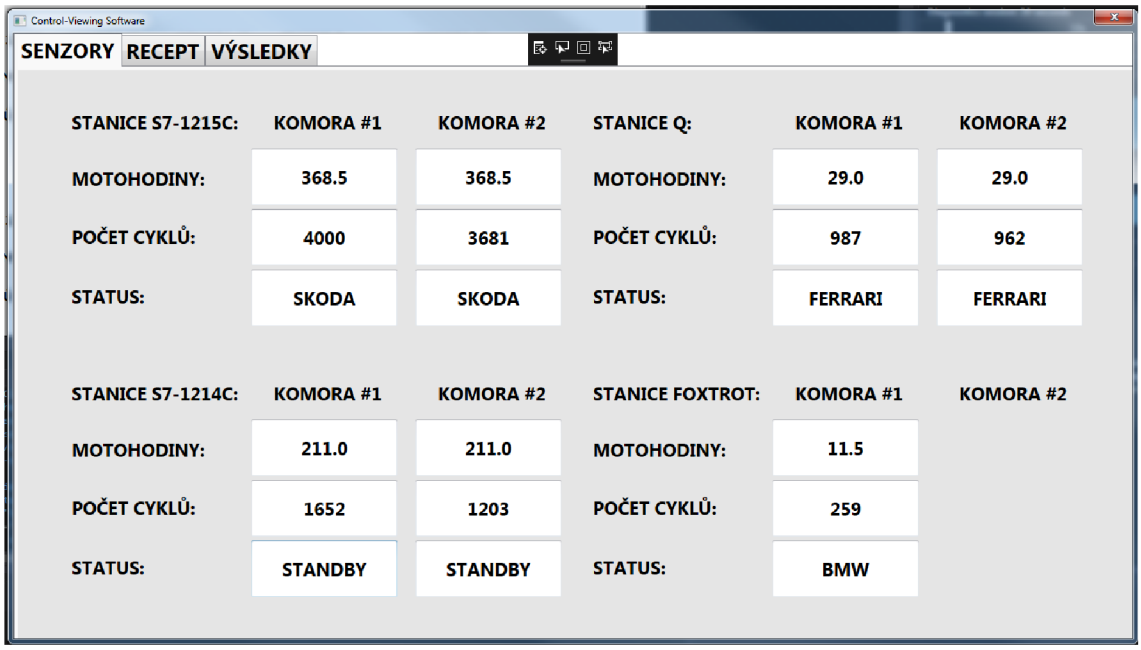
noduchost a modernost zvolen objektový programovací jazyk C# s vývojovým prostředím Visual Studio 2015. Právě tento jazyk a toto prostředí nabízejí více než dostačující prostředky pro vytvoření požadované aplikace. Tyto součásti byly zvoleny i s ohledem na požadavek třetích stran, které s ním chtějí v budoucnu začít více pracovat a aplikaci dále rozvíjet. Pro databázový systém aplikace bylo na základě stejných klíčů výběru zvoleno SQLite ve verzi č. 3.

Celá aplikace bude implementována jako vícevláknová, kdy bude následně probíhat komunikace mezi jednotlivými vlákny. Vícevláknový program by měl zajišťovat co nejplynulejší chod všech jeho částí. Jednotlivá vlákna lze pak rozdělit do dvou okruhů:

1. Zobrazovací vlákno: Pro zobrazování hodnot z PLC, generování nových hodnot pro PLC a pro práci s databázovým systémem.
2. Komunikační vlákno: Pro zajištění komunikace s testovacími zařízeními (bude vytvořeno samostatné komunikační vlákno pro každé připojené zařízení).

6.3.1 Grafické rozhraní

Grafické rozhraní bude obsahovat celkem 3 obrazovky. První část obsahuje informace o aktuálním vytížení, počtu vykonaných měřicích cyklů a o statutu (režimu stanice nebo o aktuální vybrané měřicí metodě) jednotlivých zařízení. Jedné se o obrazovku nazvanou **SENZORY** a její grafické rozvržení je zobrazeno na obrázku č. 6.1.

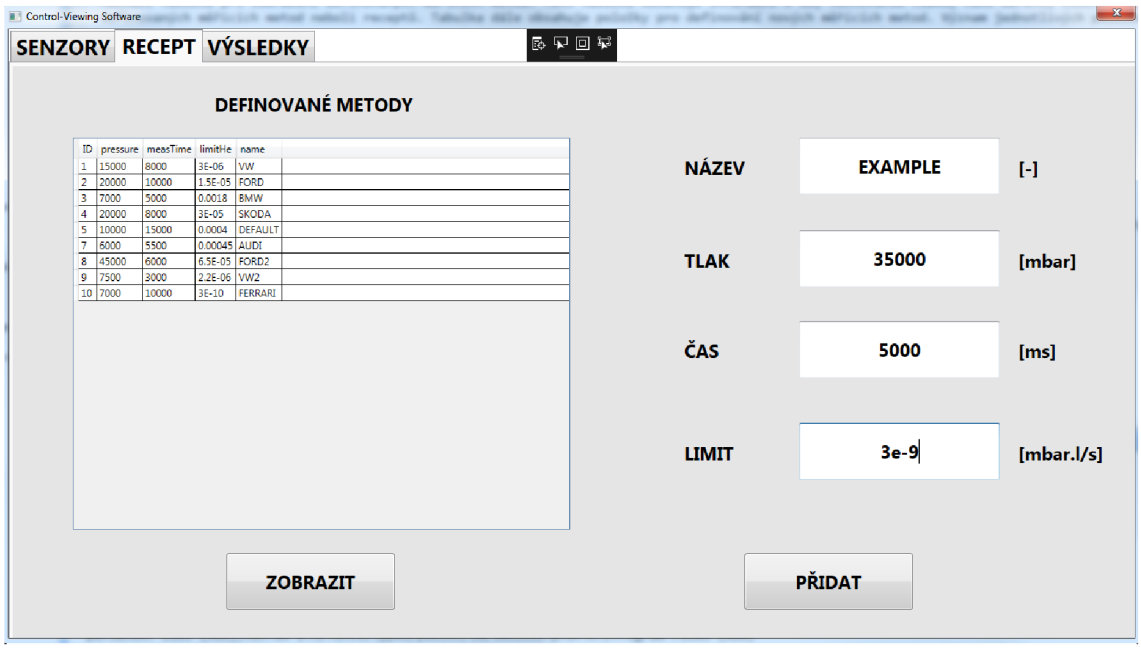


Obr. 6.1: Nadřazená aplikace, obrazovka SENZORY.

Druhou část aplikace představuje obrazovka **RECEPT**. Slouží k zobrazování a definování měřicích metod pro měřicí zařízení. Na této obrazovce se proto nachází tabulka pro zobrazení definovaných měřicích metod. Tato tabulka obsahuje informace o ID, testovacím tlaku, času měření, limitní hodnotě a názvu již definovaných měřicích metod neboli receptů. Tabulka dále obsahuje položky pro definování nových měřicích metod. Význam jednotlivých položek je následující:

1. **NÁZEV**: Jméno, pod kterým bude nová metoda uložena.
2. **TLAK**: Cílový tlak ve výrobku během měření netěsnosti (jednotka je mbar).
3. **ČAS**: Čas, po který se bude netěsnost měřit (jednotka je milisekunda).
4. **LIMIT**: Limitní hodnota netěsnosti pro vyhodnocení, zda se jedná o těsný nebo netěsný testovaný produkt (jednotka je mbar.l/s[1]).

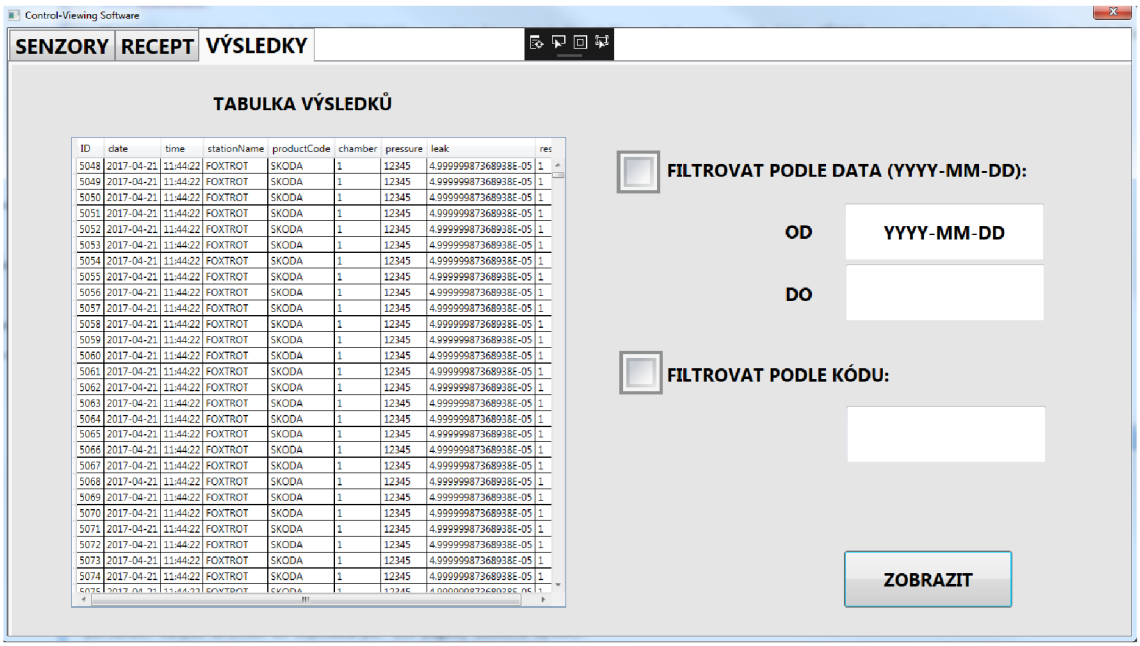
Obrazovka také obsahuje tlačítko ZOBRAZIT pro aktualizaci tabulky definovaných metod a tlačítko PŘIDAT pro zadání nového receptu do tabulky. Grafické rozvržení obrazovky **RECEPT** je zobrazeno na obrázku č. 6.2:



Obr. 6.2: Nadřazená aplikace, obrazovka RECEPT.

Poslední část aplikace představuje obrazovka **VÝSLEDKY**, viz obrázek č 6.3. Tato obrazovek slouží pro zobrazování zaznamenaných výsledků ze všech měřicích stanic v systému chytré továrny. Tabulka na této obrazovce poskytuje informace o ID, datu a času, identifikátoru stanice, identifikátoru měřeného produktu, číslu

[1] 1 mbar.l/s znamená, že za normálních pokojových podmínek dochází ke ztrátě 1 krychlového centimetru plynu za sekundu.



Obr. 6.3: Nadřazená aplikace, obrazovka VÝSLEDKY.

komory stanice, výsledném tlaku ve výrobku, hodnotě netěsnosti a o konečném vyhodnocení (0 = chyba během testování, 1 = OK, 2 = NOK) jednotlivých vykonaných testů. Obrazovka umožňuje i filtrování výsledků, a to jednak podle zvoleného časového rozmezí (mezi dvěma daty) a také podle kódu testovaného produktu.

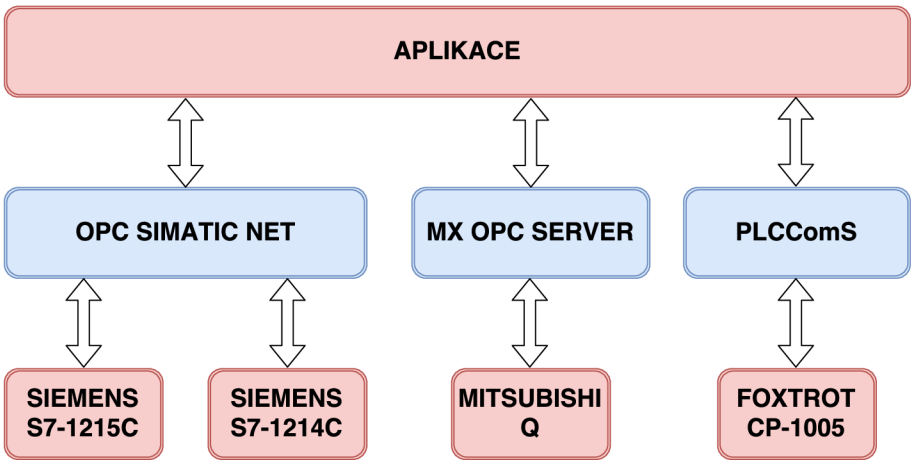
6.3.2 Komunikace s jednotlivými zařízeními

Aplikace je určena pro desktopový počítač a bude zajišťovat komunikaci se čtyřmi měřicími zařízeními řízenými PLC systémy. Jak již bylo výše zmíněno, jedná se o tyto PLC: 2xSiemens S7-1200, 1x Mitsubishi Q26 a 1x Teco Foxtrot CP-1005. Aby byla aplikace vhodná pro využití v průmyslovém prostředí, je nutné, s ohledem na toto prostředí, zvolit i komunikační nástroje pro komunikaci s jednotlivými zařízeními. Kvůli této potřebě bude aplikace pro komunikaci s PLC systémy využívat příslušné OPC servery, protože tento model komunikace PLC <-> OPC <-> APLIKACE je přímo určen pro vizualizaci a řízení procesů v průmyslovém prostředí. Na počítači s aplikací proto z těchto důvodů budou nainstalovány i následující OPC servery:

1. Siemens: Simatic NET V12.
2. Mitsubishi: MELSEC MX OPC V6.07.
3. Teco: PLCComS V4.5.

Aplikace bude koncipována jako klient jednotlivých OPC serverů, které budou zprostředkovávat komunikaci s jednotlivými systémy PLC, které řídí činnost mě-

řicích stanic. Celý princip komunikace lze pak prezentovat následujícím blokovým schématem 6.4:



Obr. 6.4: Koncept komunikace navržené aplikace s PLC systémy s využitím OPC serverů.

6.3.3 Databázový systém aplikace

Protože má aplikace zaznamenávat měřicí metody a výsledky měření, bude databázový systém obsahovat především tyto 2 tabulky. Struktura jednotlivých tabulek odpovídá klíči: Identifikační číslo (číslo pořadí) záznamu, hodnota 1, hodnota 2,..., hodnota N, aktivnost záznamu.

Tabulka pro měřící metody musí obsahovat všechny položky, které jasně definují měřicí metodu tak, aby poskytovala měřicímu zařízení všechny potřebné informace. Bude proto obsahovat následující sloupce:

1. Identifikační číslo
2. Název receptu
3. Tlak ve výrobku během testu
4. Čas testu
5. Rozhodovací limitní hodnota testu
6. Informace o tom, zda je záznam aktivní

Tabulka, do které se budou ukládat výsledky jednotlivých testů všech stanic, musí poskytovat všechny potřebné údaje pro technology nebo koncové zákazníky, kteří potřebují s informacemi dále pracovat. Bude proto obsahovat tyto sloupce:

1. Identifikační číslo
2. Datum
3. Čas

4. Název zařízení
5. Číslo testovací komory
6. Kód otestovaného výrobku
7. Výsledný tlak ve výrobku
8. Výsledná netěsnost výrobku
9. Vyhodnocení testu
10. Informace o tom, zda je záznam aktivní

Součástí databáze bude i tabulka pro zálohování motohodin, počtu cyklů a počítadel OK kusů pro jednotlivé komory všech testovacích stanic. Tato tabulka slouží jako záloha těchto dat pro případ ztráty retenční paměti[2] PLC systémů. Struktura této tabulky je proto následující:

1. Identifikační číslo
2. Název stanice
3. Číslo komory
4. Hodnota motohodin při záloze
5. Celkový počet vykonaných testů - zaznamenáváno v čase zálohování
6. Celkový počet testů, které byly vyhodnoceny jak OK - zaznamenáváno v čase zálohování
7. Informace o tom, zda je záznam aktivní

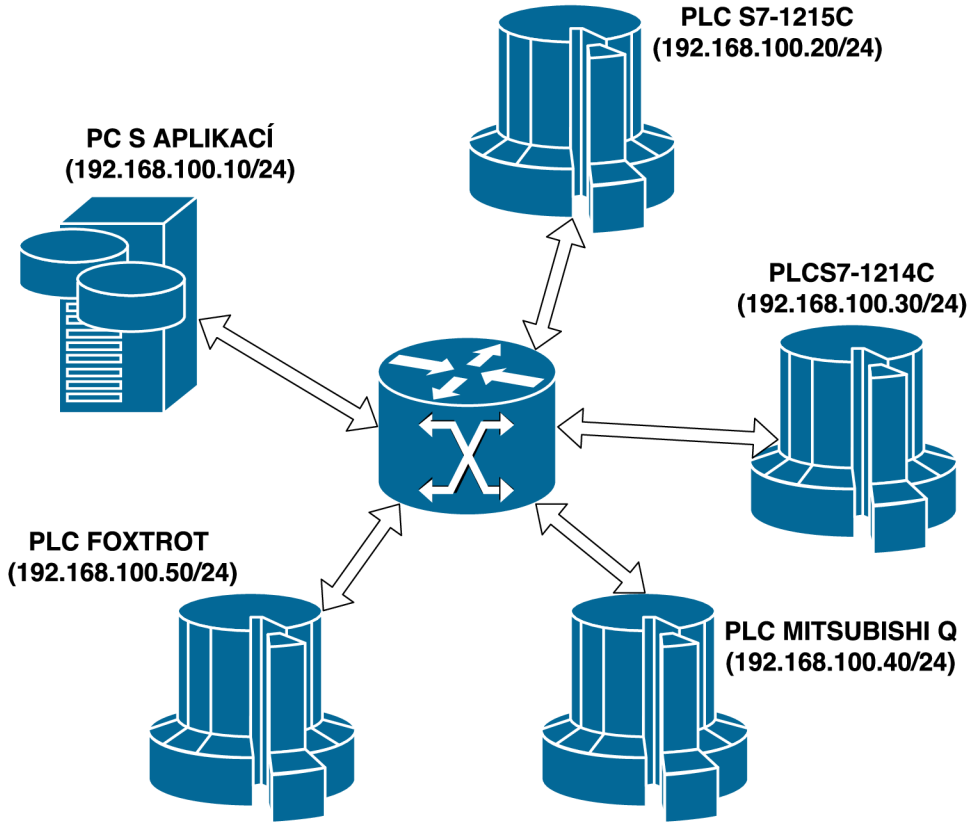
6.4 Topologie komunikační sítě v továrně

Se všemi uvedenými zařízením lze komunikovat s využitím klasické Ethernetové sítě, i když se jednotlivé komunikační protokoly navzájem nějakým způsobem liší, viz kapitola 4.3 o nativních komunikačních protokolech. Vzhledem k rozmístění všech zařízení je zvolena hvězdicová topologie sítě, kdy jednotlivá zařízení budou připojena do sítě pomocí klasického přepínače. Ten umožňuje přepínání rámců od všech připojených zařízení, protože všechny komunikační protokoly právě klasických ethernetových rámců pro přenos dat využívají. Tato topologie je zvolena také z důvodu snadného přidání nově vyrobených měřicích stanic do systému. Pro připojená zařízení je v továrně vyhrazen následující rozsah IP adres: 192.168.100.0/24. Vytvořená topologie je znázorněna na obrázku č. 6.5.

6.5 Komunikační protokol v chytré továrně

Komunikační protokol je prakticky vyřešen volbou topologie hvězdy založené na technologii Ethernet a také vhodným naprogramováním uživatelské aplikace. Všechny

[2]Retenční paměť slouží k dlouhodobé paměti dat, je napájena baterií nebo kondenzátorem.

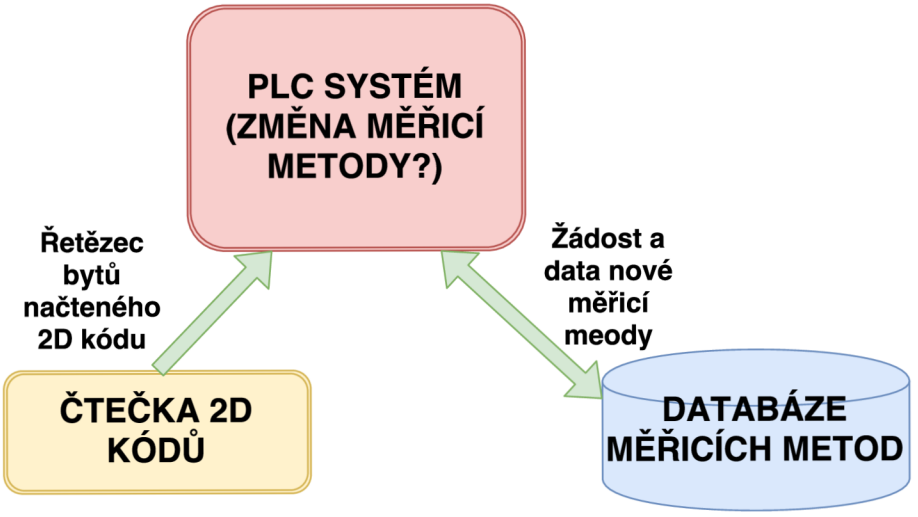


Obr. 6.5: Topologie navržené komunikační sítě v chytré továrně.

4 zařízení v topologii chytré továrny budou komunikovat pouze s jedním zařízením, v tomto textu vedeným jako nadřazený systém. V tomto nadřazeném systému jsou všechny softwarové aplikace (především uživatelská aplikace a příslušné OPC servery), které budou komunikaci iniciovat a výměnu dat také řídit. Vhodným nastavením některých OPC serverových aplikací pro komunikaci s PLC systémy lze docílit co nejmenšího zatížení komunikační sítě datovým provozem. Pokud nejsou takové možnosti v OPC, budou doplněny do komunikačního vlákna uživatelské aplikace v nadřazeném systému. Pro co nejmenší zatížení komunikační sítě a pro plynulý chod aplikace je nutné komunikaci naprogramovat tak, aby došlo k přenosu proměnných pouze při jejich změně. Tím se zamezí neustálého přenosu proměnných, jejichž změna není častá.

Komunikačním protokolem, v kontextu s touto prací, je tedy myšlen především způsob automatického vybírání měřicích metod jednotlivých zařízení tak, aby bylo zajištěno bezpečného a plynulého chodu měřicích stanic a nadřazené aplikace. Do PLC systému bude nutné doplnění algoritmu, který bude vybírat měřicí metodu pouze tehdy, bude-li to opravdu potřebovat (pouze při zjištěné změně typu testovaného výrobku) a nebude docházet k přenosu parametrů měřicích metod s každým načte-

ným kódem v testovacím zařízení. Při výběru a implementaci nové měřicí metody je také nutné brát v úvahu skutečnost, že se metoda nesmí změnit během vlastního měření netěsnosti výrobku. To by fatálním způsobem ovlivňovalo kvalitu celé výroby v chytré továrně a finálních výrobků.



Obr. 6.6: Schéma principu výběru měřicí metody.

7 POPIS IMPLEMENTACE SYSTÉMU URČENÉHO PRO CHYTROU TOVÁRNU

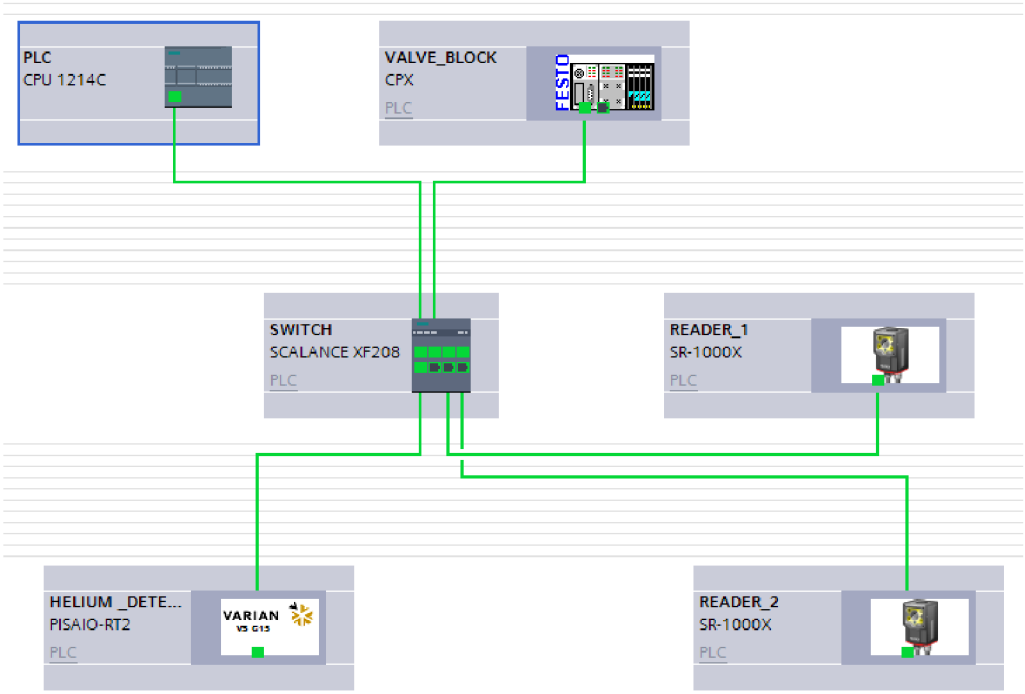
7.1 Konfigurace zařízení v topologii chytré továrny

V této kapitole je popsána konfigurace měřicích zařízení, která souvisí přímo s touto prací. Z tohoto důvodu bude prezentována implementace čtečky 2D kódů do PLC Siemens S7-1214C a na stejném zařízení je prezentována i následná úprava softwaru. Konfigurace a úprava softwaru na ostatním PLC systémech je prakticky stejná.

7.1.1 Implementace čtečky 2D kódů

Pro čtení 2D kódů, které identifikují výrobky, je každá měřicí komora doplněna o čtečky Keyence SR-1000 (tento typ je použit s PLC systémy Siemens a Mitsubishi, s PLC Teco je použit typ Datalogix Matrix 210N). Tyto čtečky budou snímat identifikační kódy výrobků založených právě do měřicích komor. Čtečky podporují mnoho komunikačních protokolů a je možné je uzpůsobit použitému PLC. Pro spojení s PLC Siemens je využit Profinet. Čtečka je připojena Ethernetovým kabelem do přepínače uvnitř stanice, do kterého jsou také připojena další zařízení komunikující přes Profinet. Zapojení vnitřní Profinetové sítě stanice je zobrazeno na obrázku č. 7.1.

Na obrázku č. 7.1 je znázorněna hardwarová konfigurace měřicí stanice. Kromě vlastního PLC Siemens S7-1214C a čteček Keyence SR-1000 je Profinetová síť využita i pro komunikaci s ventilovým blokem a s vlastním hmotnostním spektrometrem pro měření hélia. Konfigurace samotných čteček (stejně jako všech ostatních zařízení v této síti) je pak provedena pomocí GSD (General Station Descrition) souborů, které jsou součástí dokumentace k zařízením. GSD soubory popisují zařízení pro implementaci do Profinetové sítě. Zařízením jsou po přidání do sítě přiřazeny adresy, které PLC následně používá pro vzájemnou komunikaci. Všechna zařízení připojená Profinetem k PLC se pak chovají jako jedno autonomní zařízení (autonomní stroj). U dalších PLC systémů není použit Profinet, ale protokol MELSEC pro komunikaci s Mitsubishi a protokol TCP pro komunikaci s Teco. Pro ovládání čteček je v PLC naprogramován funkční blok, který obsluhuje načítání a zpracování 2D kódů. Funkční blok pro komunikaci se čtečkou a načítání 2D kódů je nazván B01_ReadProductCode.



Obr. 7.1: Konfigurace sítě Profinet v PLC Siemens S7-1215C.

7.1.2 Úprava softwaru v PLC

Struktura softwaru implementovaného do PLC Siemens S7-1214C je rozdělena do několika bloků. Do PLC bylo doplněno čtení 2D kódů, výběr měřicí metody a čítače vytížení. Ostatní části jsou součástí původního kódu. Struktura kódu po provedených úpravách vypadá následovně:

1. Čtení 2D kódů.
2. Výběr nové měřicí metody nebo ponechání stávající.
3. Automatický měřicí cyklus.
4. Odeslání výsledků do databáze.
5. Detekce stavů a chyb měřicího stroje.

Do PLC byly doplněny datové bloky pro dosažení požadované struktury pro proměnné měřicí metody (blok Method), výsledků (blok Results) a čítačů zatížení (blok Load).

Ve funkčním bloku B02_SelectMeasurementMethod určeném pro výběr nové měřicí metody je implementován program, který po načtení kódu z výrobku najde klíčové znaky v přečteném kódu a porovná je s klíčovými znaky v aktuální měřicí metodě, podle které stanice měří. Po vyhodnocení rovnosti nebo nerovnosti těchto znaků pak PLC případně zažádá nadřazenou aplikaci o novou měřicí metodu. Do nadřazené aplikace je z PLC odesílán i kód výrobku. Tento kód výrobku je ná-

Obr. 7.2: Úprava softwaru v PLC: Přidané nebo upravené datové bloky.

sledně, kromě identifikace výrobku v databázi výsledků, využíván v aplikaci právě pro načtení příslušné měřicí metody z databáze. Aplikace následně odešle vybranou měřicí metodu do PLC. Kód pro vyhodnocení výběru nové metody a odeslání žádosti o metodu je uveden zde (proměnné uvedené v uvozovkách představují název datového bloku s proměnnými, číslo v hranatých závorkách vyjadřuje číslo měřicí komory):

```
// Vyhodnoceno jako TRUE, pokud je dokončena funkce načítání 2D kódu
IF "StepVariables".ReadingCode[1].Q THEN
  // Hledání charakteristických znaků pro výběr metody
  IF (FIND(IN1 := "Results".Results[1].ProductCode, IN2 :=
      "Method".MET[1].K_SET_MethodName) = 0) THEN
  // Odeslání žádosti o metodu
  "Method".MET[1].REQUEST := TRUE;
  END_IF;
END_IF;
```

Součástí úpravy softwaru v PLC je i implementace čítačů vytížení měřicí stanice. Tyto čítače jsou doplněny do funkčního bloku pro detekci stavů a chyb, který je v softwaru nazván B05_StateAndErrorDetection. Přičítání motohodin se provádí v půlhodinových intervalech. To zajišťuje požadovanou přesnost. Čítače jsou implementovány následujícím způsobem:

```
// Časovač pro počítání motohodin
#TON1(IN := TRUE, PT := T#30m);

// Po uplynutí definovaného času dojde k přičtení tohoto časového úseku k
    hodnotě počítadla motohodin
IF #TON1.Q THEN
  "Load".Hours[1] := "Load".Hours[1] + 0.5;
  #TON1(IN := FALSE, PT := T#30m);
  RESET_TIMER(#TON1);
END_IF;
```

Posledním doplněním do softwaru PLC je zpracování výsledků. Pro vložení výsledků do správného tvaru (do databloku Results) a následné nastavení požadavku na zápis do databáze slouží funkční blok B04_SendResultsToDatabase. Implementace je provedena i v ostatních PLC systémech (Mitsubishi i Teco) prakticky stejně

jako u PLC Siemens, které je prezentováno výše. Proto by byl jejich popis redundantní. Rozdíly v jednotlivých PLC zařízeních jsou pouze v syntaxi strukturovaného programovacího jazyku a v použití proprietárních síťových protokolech pro komunikaci. Zde popsaný software pro PLC Siemens S7-1214C (vytvořený v TIA Portal Basic V13) je obsažen na přiloženém CD.

7.1.3 Nastavení OPC serverů

Všechny OPC servery jsou nastaveny podle příslušných manuálů. Popis nastavení jednotlivých OPC serverů by byl obsahově velmi náročný a příslušné konfigurační soubory jsou pro jednotlivé OPC severy na přiloženém CD. Na přiloženém CD se nachází i příslušné návody na konfiguraci. Z tohoto důvodu je zde uveden pouze přehled OPC serverů s odkazem na příslušné dokumenty s popisem konfigurace:

1. **Siemens**: Simatic NET V12, nastavení popsáno v [19].
2. **Mitsubishi**: MX OPC Server V6.0.7, nastavení popsáno v [20].
3. **Teco**: PLCComS V4.5, nastavení popsáno v [21].

7.2 Funkce aplikace pro chytrou továrnu

V této části textu jsou podrobně prezentovány principy funkcí vytvořené aplikace. Jednak bude popsána komunikace s jednotlivými PLC systémy, která po vytvoření hvězdicové Ethernetové sítě tvoří síť pro centralizovaný sběr dat. Dále bude také popsána práce aplikace s databází a nakonec se bude tato kapitola věnovat principu zobrazování hodnot vytížení a parametrů v aplikaci. Grafická podoba aplikace přesně odpovídá popisu v návrhu aplikace, kapitola 6.3.1.

Jak již bylo zmíněno v návrhu, celá aplikace je napsána v objektovém programovacím jazyce C#. Je koncipována jako vícevláknová a vlákna jsou rozdělena na zobrazovací a komunikační. Pro zajištění plynulého chodu aplikace je proto vytvořeno celkem pět programových vláken:

1. 1. vlákno: Pro zobrazování hodnot na příslušných obrazovkách, generování nových hodnot určených k zápisu do PLC a pro obsluhu databázového systému.
2. 2. vlákno: Obsluha komunikace s měřicí stanicí řízené systémem PLC S7-1215C.
3. 3. vlákno: Obsluha komunikace s měřicí stanicí řízené systémem PLC S7-1214C.
4. 4. vlákno: Obsluha komunikace s měřicí stanicí řízené systémem PLC Mitsubishi Q26.
5. 5. vlákno: Obsluha komunikace s měřicí stanicí řízené systémem PLC Foxtrot CP-1005.

Po spuštění aplikace nejprve metoda MainWindow() z hlavní třídy MainWindow zajišťuje vytvoření instance třídy Database. Tato třída je následně využívána pro vytvoření databázového systému, přístup do databázového systému a práci s ním:

```
Database database = new Database();
```

Při vykonávání programu po jeho spuštění následují instance jednotlivých tříd pro komunikaci s PLC systémy. Jsou vytvořena jednotlivá vlákna, která zajišťují vytvoření připojení k OPC serverům a následně řídí komunikaci. Komunikační vlákna potom řídí jak příjem dat z PLC do aplikace, tak i odesílání dat z aplikace do PLC. Instance tříd a vytvoření komunikačních vláken je prezentováno zde:

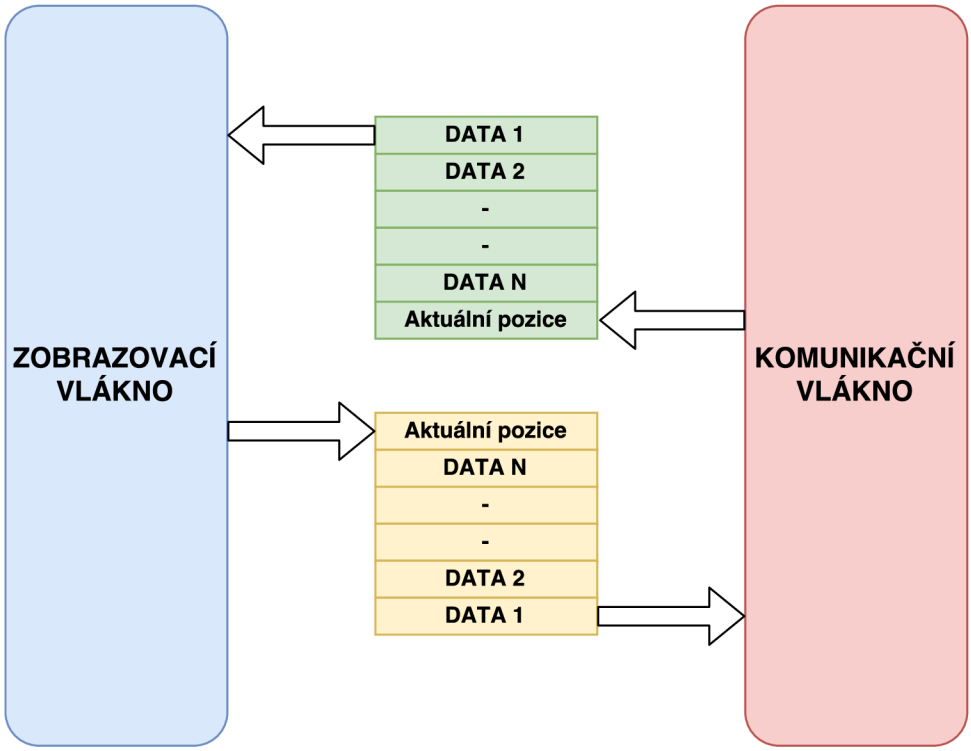
```
// Pro komunikaci s PLC Teco Foxtrot CP-1005
CommunicationTECO plcComS = new CommunicationTECO();
Thread CommunicationPLCComS = new Thread(new ThreadStart(()=>
    plcComS.ConnectToServer("127.0.0.1", 5010)));
CommunicationPLCComS.Start();

// Pro komunikaci s PLC Siemens S7-1215C a PLC S7-1214C
CommunicationSIEMENS simaticNet = new CommunicationSIEMENS();
simaticNet.initOPCVariables();
Thread CommunicationSimaticNET = new Thread(new ThreadStart(()=>
    simaticNet.ConnectToServer("opc://localhost/OPC.SimaticNET")));
CommunicationSimaticNET.Start();

// Pro komunikaci s PLC Mitsubishi Q26
CommunicationMITSUBISHI mxOpc = new CommunicationMITSUBISHI();
mxOpc.initOPCVariables();
Thread CommMXOPC = new Thread(new ThreadStart(()=>
    simaticNet.ConnectToServer("opc://localhost/Mitsubishi.MXOPC")));
CommMXOPC.Start();
```

Vždy je vytvořeno vlákno, které volá metodu pro vytvoření připojení k OPC server. Tato metoda má jeden vstupní parametr, a tím je vždy adresa příslušného OPC severu. OPC server je definován buď tzv. socketem, jako je tomu u PLCComS pro PLC Teco Foxtrot (IP adresa a číslo portu), nebo url adresou u OPC serverů Simatic NET pro PLC Siemens a MXOPC pro PLC Mitsubishi. Protože jsou všechny OPC servery na stejném počítači jako nadřazená aplikace, využívá se vždy lokální IP adresy. U systémů Siemens a Mitsubishi je vždy po instanci třídy zavolána metoda initOPCVariables(), která zajišťuje registraci potřebných proměnných pro výměnu dat. Tato metoda je pro každé PLC upravená podle způsobu práce s příslušným OPC serverem. Posledním krokem vykonaným po těsně spuštění aplikace je vytvoření zobrazovacího vlákna:

```
Thread SetPanels = new Thread(new ThreadStart(SetPanelFromQueue));
SetPanels.Start();
```

Při vytvoření zobrazovacího vlákna je volána metoda SetPanelFromQueue(). Ta zajišťuje veškeré zobrazování a generování hodnot a obsluhuje databázový systém. Všechna vytvořená vlákna se po spuštění pořád udržují „živá“. To znamená, že do ukončení aplikace nedochází k jejich ukončení.

Pro komunikaci mezi jednotlivými vlákny komunikačními vlákny a zobrazovacím vláknem je využito tzv. thread-safe bufferu (bezpečného pro přístup z více vláken) typu fronta neboli přesně FIFO - First In First Out. Tato paměť je použita 2x: Jednou je definována pro příjem dat z PLC do aplikace, podruhé pro odesílání dat z aplikace do PLC. Princip spolupráce komunikačního a zobrazovacího vlákna je zobrazen na obrázku č. 7.3.



Obr. 7.3: Vzájemná komunikace vláken programu.

Oba dva buffery jsou definovány ve třídě Communication. Popis této třídy je uveden níže. Buffery pracují se strukturou TVARIABLE, která obsahuje všechny položky nutné pro vzájemnou komunikaci jednotlivých vláken. Ve struktuře TVARIABLE je definován identifikátor proměnné, hodnota proměnné (číslo nebo textový řetězec) a případně indexy, pokud se jedná u jednorozměrné nebo dvourozměrné pole. Struktura proměnných pro mezivláknovou komunikaci je definována ve statické třídě Definitions.

```
// Struktura proměnných pro mezivláknovou komunikaci
public struct TVARIABLE
{
  public int num;       // Identifikátor
  public string var;    // Text, pokud se jedná o řetězec
  public int x;         // Index pole x
  public int y;         // Index pole y
  public double value;  // Hodnota, pokud se jedná o číslo
}

// Buffery pro mezivláknovou komunikaci
public ConcurrentQueue<TVARIABLE> FIFOFromPLC = new
    ConcurrentQueue<TVARIABLE>();
public ConcurrentQueue<TVARIABLE> FIFOToPLC = new
    ConcurrentQueue<TVARIABLE>();
```

Pro vytvoření bufferů pro mezivláknovou komunikaci je využita nativní knihovna Visual Studia 2015, konkrétně pak třída ConcurrentQueue<T>, kde parametr T vyjadřuje datový typ prvku v bufferu.

7.2.1 Komunikace aplikace s PLC systémy

Pro komunikaci aplikace s PLC systémy s využitím OPC jsou v programu vytvořeny 3 třídy, každá pro jiný typ systému PLC (1 třída pro obě PLC Siemens, 1 třída pro PLC Mitsubishi a 1 třída pro PLC Teco). Typ síťové komunikace vždy závisí na použitém OPC serveru a jeho definovaném rozhraní pro komunikaci s ním. Je proto vytvořena třída pro komunikaci s PLC systémy Siemens, třída pro komunikaci s PLC systémem Mitsubishi a třída pro komunikaci se systémem Teco. Nejdůležitější prvky těchto tříd jsou popsány v této části textu.

U každého PLC systému platí skutečnost, že proměnnou, kterou chceme monitorovat nebo do ní zapisovat nové hodnoty, musíme „zaregistrovat“. Každá proměnná je v PLC definována určitou adresou. Tato adresa je právě používaná pro „registraci“ proměnných. Pokud se tedy aplikace OPC serveru dotazuje na nějakou proměnou, musí ji definovat příslušnou adresou. Ke každé adrese je softwarově přiřazen i tzv. index proměnné. Tento index je pak využíván pro identifikaci proměnných v aplikaci.

Připojení k PLC Siemens S7-1200

Pro připojení k PLC systémům Siemens byla vytvořena třída CommunicationSIEMENS, která dědí ze třídy Communication. Ve třídě CommunicationSiemens je oproti třídě Communication navíc metoda initOPCVariables(). Tato metoda definuje proměnné a jejich adresy, se kterými pak bude aplikace pracovat a používat je při komunikaci s OPC serverem. Pro registraci proměnných PLC Siemens byla

vytvořena struktura PLCVariables, která obsahuje adresu proměnné a informaci o stavu proměnné (aktivní nebo neaktivní):

```
// Struktura pro definici proměnných
public struct PLCVariables
{
  public int access;     // Stav proměnné.
  public string address; // Adresa proměnné
}
```

Metodou initOPCVariables() je vytvořeno pole struktur PLCVariables, kde jednotlivé položky definují proměnné. Kromě položek ve struktuře PLCVariables se pro identifikaci proměnné využívá i index struktury v poli struktur:

```
public void initOPCVariables()
{
  plcVariables[Definitions.RESULTS + 0].address =
      "S7:[S7_Connection_1]DB19,REAL0";
  plcVariables[Definitions.RESULTS + 0].access = 1;

  plcVariables[Definitions.RESULTS + 10].address =
      "S7:[S7_Connection_2]DB19,REAL0";
  plcVariables[Definitions.RESULTS + 10].access = 1;
}
```

Proměnná address obsahuje řetězec s adresou proměnné (identifikuje datový blok, typ a ofset proměnné v PLC). Proměnná access vyjadřuje, zda je proměnná aktivní nebo není (1 = ano, -1 = ne). Všechny hodnoty indexů jsou přiřazeny určitým skupinám proměnných. V uvedeném příkladu je index uveden jako RESULTS + X z třídy Definitions, kde RESULTS vyjadřuje nějaké číslo definující výsledek a X vyjadřuje posunutí od první hodnoty v daném rozsahu hodnot indexů.

Ve třídě Communication jsou vytvořeny metody, které zajišťují připojení k OPC serveru, odesílání dat do PLC, příjem dat z PLC pouze při změně hodnoty. Vytvořené metody jsou s krátkými komentáři uvedeny zde:

```
public class Communication
{
  // Připojení k serveru a uderžování spojení.
  public void ConnectToServer(string serverUrl);

  // Odpojení od serveru při ukončení aplikace
  public void DisconnectFromServer();

  // Přečtení hodnoty z mezivláknové komunikace a odeslání do PLC
  private void WriteToPLC();
```

```
    // Čtení hodnot z PLC při jejich změně a jejich zápis do mezivláknové
        komunikace
    private void ReadAutoUpdate(IList<DataValue> DataValues);

    // Registrace proměnných definovaných v poli struktur plcVariables k
        monitorování jejich hodnot
    private void MonitorOPCVariables();

    // Zastavení monitorování hodnot při ukončení aplikace
    private void stopMonitorItems();
}
```

Třída Communication vychází z příkladu komunikace OPC serveru Simatic NET s klientem napsaným v jazyce C#, příklad byl převzat z [18].

Připojení k PLC Mitsubishi Q26

Pro připojení k PLC systému Mistubishi byla vytvořena třída CommunicationMITSUBISHI, která opět dědí ze třídy Communication. Je opět napsána metoda initOPCVariables(), které pracuje se strukturou plcVariables, kde se definují adresy pro přístup k proměnným. Princip definice proměnných pro komunikaci je podobný jako pro PLC systémy Siemens. Ukázka definice proměnných pro komunikaci s OPC serverem Mitsubishi ve třídě CommunicationMITSUBISHI:

```
public void initOPCVariables()
{
  plcVariables[Definitions.RESULTS + 20].address = "D9_Results.OK_1";
  plcVariables[Definitions.RESULTS + 20].access = 1;

  plcVariables[Definitions.RESULTS + 30].address = "D9_Results.OK_2";
  plcVariables[Definitions.RESULTS + 30].access = 1;
}
```

Položka address obsahuje řetězec s názvem položky, která je vytvořená a asociovaná s příslušnou adresou v MX OPC serveru. Pro komunikaci aplikace s MXOPC Mitsubishi jsou použity stejné funkce jako pro komunikaci Simatic NET Siemens. S oběma OPC servery se pracuje stejným způsobem.

Připojení k PLC Teco Foxtrot CP-1005

Pro komunikaci s PLC Teco Foxtrot je vytvořena třída CommunicationTECO, která se výrazně liší od tříd pro komunikaci s OPC Simatic NET Siemens a MXOPC Mitsubishi uvedených výše. OPC server PLCComS pro Teco Foxtrot pracuje jako TCP server, který naslouchá na portu 5010 (výchozí hodnota, možnost konfigurace na jiný nebo přidání dalšího portu pro další PLC). Aplikace (TCP klient) mu po

vytvoření spojení pošle sérii příkazů, kterými OPC serveru sdělí, jakým způsobem s ním chce pracovat. Následně pouze naslouchá na svém přesněji nedefinovaném portu informace o hodnotách proměnných. Zároveň je port 5010 OPC serveru využívám i pro odesílání nových hodnot proměnných z aplikace do PLC. OPC server PLCComS poskytuje aplikaci vždy název proměnné a hodnotu proměnné. Aplikace vždy přijme řetězec bajtů, které musí vhodným způsobem dekódovat. Příslušné adresy proměnných jsou definovány při psaní softwaru pro PLC a následném nastavení připojení PLC s OPC serveru PLCComS. Přehled základních metod definovaných ve třídě CommunicationTECO s krátkým popisem je uveden zde:

```
public class CommunicationTECO
{
  // Vytvoří a udržuje TCP spojení s OPC serverem běžícím na definované
      adrese  portu
  public void ConnectToServer(string IP, Int16 port);

  // Přijme řetězec bajtů z bufferu komunikace a zavolá funkci pro zpracování
  private void ReadData();

  // Dekóduje přijatý řetězec bajtů, výsledkem je název přijaté proměnné a
      její hodnota
  private int decodeTCP (byte[] receivedBytes);

  // Zapisuje nové hodnoty do proměnných v PLC, vytváří řetězec s názvem
      proměnné a s novou hodnotou
  private void WriteToPLC();

  // Definuje přístupový název proměnných a jejich datový typ
  private void createPLCVarList();
}
```

Při prvním zavolání metody ConnectToServer() je do OPC serveru PLCComS odeslána skupina konfiguračních příkazů:

```
private string initCommand =
  "SETCONF:diff,no\r\n" +
  "EN:*\r\n" +
  "GET:*\r\n" +
  "SETCONF:diff,yes\r\n";

foxtrot.Send(Encoding.ASCII.GetBytes(initCommand));
```

Tyto příkazy je nutné odeslat vždy po otevření komunikace s OPC severem PLCComS. Zajišťují vzájemnou komunikaci aplikace a PLCComs. Zajišťují také nastavení komunikace tak, aby aplikace měla přístup ke všem proměnným, které jsou v PLC Teco Foxtrot CP-1005 nastaveny jako „veřejné“. Při založení komunikace

je nutné zavolat další metodu, a to createPLCVarList() taktéž ze třídy CommunicationTECO. Tato metoda opět registruje jednotlivé prvky nesoucí data (definuje adresu proměnných a jejich identifikaci indexem). Příklad registrace proměnné:

```
AddPLCVariable(Definitions.RESULTS + 63, "Vysledky.NOK",
    Definitions.typeINTBOOL);
```

Kromě výše uvedených základních metod, třída CommunicationTECO obsahuje ještě další tři metody, které úzce souvisí s naprogramovaným zpracováním a vytvářením proměnných při komunikaci. Před uvedením těchto metod bude proto představen princip získání indexu a typu z názvu a hodnoty přijímaných proměnných a naopak princip získání názvu proměnné z indexu při odesílání hodnot.

V metodě createPLCVarList() se s každou definovanou proměnnou volá další metoda, a to AddPLCVariable(int _num, string _address, int _type), do které vstupují parametry: hodnota indexu proměnné, přístupový název proměnné a typ proměnné. Tato metoda přidává proměnou definovanou vstupními parametry do hash tabulek, které jsou celkem dvě. Hash tabulky vždy obsahují klíč a hodnotu a právě klíč se využívá pro velmi rychlé vyhledání a přístup k příslušné hodnotě, viz následující tabulky:

Tab. 7.1: Příklad hash tabulek pro přijatá a odesílaná data.

Klíč	Hodnota
Název proměnné ("Vysledky.NOK")	Hodnota indexu proměnné (RESULTS + 63)
Klíč	**Hodnota**
Index proměnné (RESULTS + 63)	Název proměnné ("Vysledky.NOK")

První hash tabulka je určena pro získání indexu z přijaté proměnné, kdy přijatý řetězec s názvem proměnné slouží jako klíč k získání hodnoty indexu a informaci o datovém typu proměnné. Index se následně využívá k přiřazení hodnoty ke grafickému objektu. Informace o datovém typu se zde využívá pro příslušné formátování hodnoty proměnné. Druhá hash tabulka pracuje opačným způsobem. Ze známého indexu, který nyní představuje klíč, vrátí název proměnné. Název proměnné je pak využit pro vytvoření příkazu (ve tvaru SET: NÁZEV, HODNOTA) nesoucího novou hodnotu proměnné, který je určen k odeslání do PLCComS a následně do PLC. Zbylé tři metody třídy CommunicationTECO jsou vedeny zde:

```
public class CommunicationTECO
{
```

```
    // Vrátí hodnotu indexu na základě názvu proměnné (pracuje s příslušnou
        hash tabulkou)
    private void getNumFromString(string _address, string _value);

    // Vrátí hodnotu proměnné na základě indexu (také pracuje s příslušnou
        hash tabulkou)
    private string getVarNameFromNum(int _num);

    // Přidá příslušné prvky do hash tabulek
    private void AddPLCVariable(int _num, string _address, int _type);
}
```

7.2.2 Databázový systém

V této části textu je popsáno využití třídy Database a metod z této třídy. Aplikace po svém spuštění vytváří instanci třídy Database. Tím dochází také k automatickému vytvoření (pouze za předpokladu, že již není vytvořen) databázového systému založeném na SQLite ve verzi 3. Vytvoření databázového systému zahrnuje vytvoření souboru pmbase.db a následně v tomto souboru vytvoření příslušných tabulek pro zaznamenávané hodnoty. Tabulky jsou vytvořeny v souladu s kapitolou 6.3.3, ve které je proveden jejich návrh. Aplikace se k souboru pmbase.db připojuje a provádí zápis a čtení v příslušných tabulkách. K tomu využívá definovaných metod a SQL příkazů. Databáze umožňuje jak zaznamenávání hodnot reálného času (výsledky měření), stejně tak i hodnot, které nejsou časově kritické. Vše je realizováno právě s využitím třídy Database.

Pro práci aplikace s databází je ve třídě Database vytvořeno několik metod, které zajišťují čtení z databáze a zápis do databáze. Níže je uveden přehled jednotlivých funkcí s komentáři, které právě databázi obsluhují. Nejdůležitějšími funkcemi jsou ty pro zápis a čtení výsledků. Funkce pro čtení výsledků z databáze je přetížená. Její vstupní parametry a jejich počet závisí na požadovaném typu filtrování jednotlivých výsledků uživatelem aplikace.

```
public class Database
{
    // Metoda zajišťující vytvoření databáze a připojení do vytvořené databáze
    public int createOpenDatabase();

    // Metoda vrací, zda je nebo není aplikace k databázovému souboru připojena
    public bool isConnected();

    // Metoda, ve které jsou definovány tabulky určené k vytvoření (vytváří se
        pouze pokud neexistují)
    private int defineSQLTables(int tn);
```

```
    // Metoda zajišťující vytváření jednotlivých tabulek
    private void createTables();

    // Metoda vkládjící řádek s výsledky do tabulky výsledků v databázi
    public void insertResultsLine(RESULTS result);

    // Přetížená metoda zajišťující čtení příslušných řádků výsledků
        filtrovaných podle kódu výrobku
    public List<RESULTS> readResultsFromDb(string code);

    // Přetížená metoda zajišťující čtení příslušných řádků výsledků
        filtrovaných podle rozmezí dvou dat
    public List<RESULTS> readResultsFromDb(string from, string to);

    // Metoda vloží řádek s nově definovanou měřicí metodou do tabulky metod v
        databázi
    public void insertMethodLine(METHOD result);

    // Metoda vrátí všechny již definované měřicí metody
    public List<METHOD> readMethod(string name);
}
```

7.2.3 Zpracování dat v aplikaci

Zpracování dat lze ve vytvořené aplikaci rozdělit do 3 větví. První větev tvoří zobrazení hodnot reálného času na příslušných grafických objektech, druhou větev tvoří zaznamenávání dat do příslušných databází a třetí větev tvoří reakce na zásahy uživatele při ovládání aplikace (reakce na tlačítka apod.). V této části textu je právě zobrazování, zaznamenávání a vytváření dat popsáno.

Zobrazování proměnných

V zobrazovacím vlákně se neustále kontroluje, zda jsou v bufferu mezivláknové komunikace (buffer FIFOFromPLC, kam zapisuje komunikační vlákno) nějaká data. Pokud se v bufferu data objeví, zobrazovací vlákno je z bufferu odebere pro svoje zpracování. Protože buffer pracuje se strukturou proměnných TVARIABLE, nachází se zde i proměnná num (index proměnné). Jak již bylo zmíněno při popisu principu komunikace s jednotlivými PLC systémy, každé přijaté proměnné je přiřazen index. Tento index jednoznačně definuje, o jakou proměnou se jedná. Zobrazovací vlákno tedy podle indexu nově nastaví příslušný grafický objekt. Zobrazování proměnných zajišťuje metoda SetPanelFromQueue() ve třídě MainWindow. Ukázka z metody je uvedena zde:

```
// Vrátí TRUE pokud je vyčtena proměnná z bufferu
if (CommunicationSIEMENS.FIFOFromPLC.TryDequeue(out buffer))
{
  // Index proměnné
  switch (buffer.num)
  {
    // Pokud odpovídá Definitions.RESULTS + 0
    case Definitions.RESULTS + 0:
      // Zobrazení proměnné do grafického objektu
      this.Dispatcher.Invoke(() => {textBox.Text = buffer.var.ToString();});
      // Zápis proměnné do připravené struktury pro výsledek
      siemens1215C[1].pressure = buffer.value;
      break;
  }
}
```

Metoda TryDequeue(out buffer) přečte proměnnou z bufferu FIFOFromPLC a podle indexu proměnnou zobrazí, případně zapíše proměnou do připravené další proměnné (například do struktury výsledků).

Zaznamenávání a prezentace výsledků měření

Všechny proměnné týkající se výsledku měření jsou po příjmu postupně přidávány do proměnných struktury RESULTS. Poslední přijatou proměnou z PLC je pak následně požadavek na zápis (nastavení příslušné proměnné v PLC, kterou aplikace také monitoruje). Aplikace zpracuje příslušnou strukturu výsledků a je zavolána metoda insertResultsLine() ze třídy Database pro zápis výsledku. Tento postup je znázorněn v následující ukázce programu:

```
// Struktura pro přijaté výsledky
public struct RESULTS
{
  public Int64 ID { get; set; }
  public string date { get; set; }
  public string time { get; set; }
  public string stationName { get; set; }
  public string productCode { get; set; }
  public Int16 chamber { get; set; }
  public double pressure {get;set;}
  public double leak { get; set; }
  public Int16 result { get; set; }
}

// Instance struktury (siemens1215C[1] pro 1. komoru, siemens1215C[1] pro 2.
    komoru měřicí stanice)
public RESULTS[] siemens1215C = new RESULTS[3];
```

```
// Zavolání metody zápisu řádku do databáze výsledku po příjmu požadavku v
    zobrazovacím vlákně)
database.insertResultsLine(siemens1215C[1]);
```

Při čtení výsledků z databáze je po požadavku na zobrazení (stisk tlačítka ZOBRAZIT na obrazovce MĚŘENÍ) zavolána funkce:

```
private void zobrazVys_Click(object sender, RoutedEventArgs e)
{
  string parameter = null;
  string from = null;
  string to = null;
  if (checkBox_datum.IsChecked.Value)
    {
      from = System.Convert.ToString(datumFrom.Text);
      to = System.Convert.ToString(datumTo.Text);

      // Načtení a zobrazení řádků výsledků podle zadaných parametrů
      tabulkaVysl.ItemsSource = database.readResultsFromDb(from, to);
    }
  else
    {
      if (checkBox_kod.IsChecked.Value)
      parameter = System.Convert.ToString(kod.Text);

      // Načtení a zobrazení řádků výsledků podle zadaných parametrů
      tabulkaVysl.ItemsSource = database.readResultsFromDb(parameter);
    }
}
```

Tato funkce podle nastavení filtrace výsledků (zaškrtnutí checkboxu v aplikaci) vyčte z databáze všechny řádky odpovídající vstupním parametrům. Načtené řádky výsledků ihned zobrazí v příslušné tabulce. Zobrazování a definice nových měřicích metod na obrazovce METODA pracuje stejným způsobem. Pro prezentaci funkce a rychlosti zobrazování výsledků z databáze je v databázi tabulka s výsledky, která obsahuje více jak 5 tisíc záznamů.

Zápis nových dat do PLC

Proměnných určených k odeslání do PLC je podstatně méně než přijímaných. Po vytvoření události v aplikaci (například stisk tlačítka nebo žádost PLC o novou metodu) se v zobrazovacím vláknu programu vloží příslušné proměnné určené k odeslání do bufferu FIFOToPLC. Komunikační vlákno si proměnnou následně přečte a odešle ji do PLC. Identifikace proměnných se provádí opět pomocí indexu proměnné

ve struktuře TVARIABLE. V zobrazovacím vláknu musí programátor všechny proměnné určené k odeslání v kódu správně identifikovat a přiřadit jim hodnotu. Program je napsán tak, že odesílání proměnných do PLC má vyšší prioritu před přijímáním. To z důvodu, aby měla obsluha aplikace okamžitou odezvu.

8 ZÁVĚR

Hlavním cílem této práce byl návrh a realizace systému, který bude zajišťovat centralizovaný sběr dat z průmyslových měřicích strojů řízených programovatelnými logickými kontroléry (PLC) s využitím Ethernetové sítě. Protože měl být systém realizován s ohledem na koncept Průmysl 4.0, tento koncept a jeho základy byly v této práci podrobně nastudovány a popsány.

Hlavním výstupem této práce je aplikace navržená jako centrální nadřazený systém. Tento systém je schopen využít Ethernetovou síť pro komunikaci s měřicími zařízeními a zároveň je schopen měřicím zařízením poskytovat požadované služby. Vytvořená aplikace umožňuje měřicím zařízením vybírat potřebné měřicí metody, dále zaznamenává a zobrazuje informace získané z měřicích zařízení. Pro zaznamenávání dat aplikace pracuje s databázovým systémem založeným na SQLite. Aplikace také představuje grafické rozhraní pro uživatele. Toto rozhraní poskytuje přehled výsledků již provedených měření, přehled měřicích metod a možnost jejich definice a také přehled o vytížení všech připojených zařízení. Protože měl být zdrojový kód aplikace jednoduše čitelný a mělo být využito moderních prostředků, byla aplikace naprogramována v objektovém programovacím jazyce C#.

Jelikož jsou měřicí zařízení řízeny PLC systémy, jsou pro komunikaci využity jejich nativní OPC servery. OPC servery byly pro vytvoření komunikace upřednostněny před různými volnými knihovnami (například S7.NET pro komunikaci s PLC Siemens S7-1200 nebo S7-1500). To z důvodů, protože jsou jednak určeny do průmyslu, komunikace s PLC systémy je vyřešená kompletně a jejich použitím se také minimalizuje zátěž komunikační sítě. Kompletní funkčnost navrženého systému pro centralizovaný sběr dat a komunikace s měřicími zařízeními v chytré továrně byla ověřena zatím pouze s jednotlivými měřicími zařízeními. Aplikace byla otestována s PLC systémem Siemens, Mitsubishi i Teco.

Součástí systému je i úprava měřicích strojů. První úpravou průmyslových měřicích zařízení bylo doplnění senzoru, který dokáže identifikovat typ výrobku určeného ke změření. Byla proto instalována čtečka 2D kódů, která zajišťuje měřicímu stroji získat všechna data obsažena na 2D kódu na produktu. Do softwaru pro PLC systémy v měřicích zařízení byla proto doplněna čtení a zpracování 2D kódů. Druhou úpravou bylo doplnění algoritmů pro automatický výběr měřicí metody a počítadla vytížení (počítání motohodin a počtu vykonaných měřicích cyklů). Tím bylo dosaženo samostatnějšího chování strojů a zvýšení „inteligence“ měřicího stroje, alespoň z pohledu nastavování vlastních parametrů měření. Všechny úpravy byly realizovány se zachováním všech jeho původních funkcí.

Z časových důvodů navržená aplikace disponuje obyčejnou grafikou převzatou z Visual Studia 2015. Stále nebyl dokončen pokročilý grafický návrh, který by měl

být pro aplikaci určen. Z časových důvodů také ještě nebyla realizována zátěžová zkouška celého systému v továrně v reálných podmínkách. Navržený systém však bude využíván a dále rozvíjen. Především na softwaru vytvořené aplikace se bude dále pracovat. Aplikace je již nyní použitelná pro všechny požadavky dle zadání této práce a, kromě jejího využití jako nadřazeného systému pro centralizovaný sběr dat, bude sloužit také jako aplikace pro vizualizaci procesů na operátorských panelech jednotlivých měřicích zařízení. Zde nahradí zastaralé aplikace. Jednoznačnou výhodou navržené aplikace je, že je již nyní schopna pracovat se širokou škálou PLC systémů a je připravena pro implementaci komunikace s dalšími typy PLC systémů. Další výhodou aplikace je, že umožňuje rozsáhlou práci s databází. Tímto je dosaženo dlouhodobého zaznamenávání dat z měřicích zařízení.

LITERATURA

[1] VERMESAN, O.; FRIESS, P.: Internet of Thing - From Research and Innovation to Market Deployment. Denmark: River Publishers Aalborg, 2014. ISBN 978-87-93102-94-1.

[2] WHAT IS THE SIMPLE MEANING OF INTHERNET OF THINGS [online]. [cit. 2017-03-27]. Dostupné z: <https://www.quora.com/What-is-the-simple-meaning-of-Internet-of-Things>.

[3] ROSE, K.; ELDRIGE, S.; LYMAN, Ch.: Internet of Thing: An Oveview - Understanding the Issues an Challenges of a More Connected World. The Internet Society, 2015. Dostupné z: <`https://www.internetsociety.org/iot`>.

[4] ELEONORA BORGIA: The Internet of Things vision: Key features, applications and open issues. *Computer Communications* [online]. 5.11.2014 [cit. 2017-03-01]. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0140366414003168>

[5] AL-FUQAHA, A.; GUIZANI, M.; MOHAMMADI, M.; ALEDHARI, M.; AYYASH, M.: Internet of Thing: A Survey on Enabling Technologies, Protocols and Applications. *IEEE Communications Surveys & Tutorials* [online]. 5.11.2016 [cit. 2017-03-01]. Dostupné z: <http://www.researchgate.net/publication/279177017>. .

[6] LI DA XU; WU HE; SHANCANG LI: Internet of Thing in Industries: A Survey. *IEEE Transactions on Industrial Infomatics* [online]. 14.10.2014 [cit. 2017-03-01]. Dostupné z: <http://ieeexplore.ieee.org/document/6714496/>.

[7] RFC 7452: *Architectural Considerations in Smart Object Networking*. 2015. Dostupné z: <http://www.rfc-editor.org/info/rfc7452>.

[8] JAY LEE; HUNG-AN KAO; SHANHU YANG: Service innovation and smart analytics for Industry 4.0 and big data environment. *ScienceDirect* [online]. 2014 [cit. 2017-03-06]. Dostupné z: <http://www.sciencedirect.com/>.

[9] SHUH G.; POTENTE T; WESCH-POTENTE C.; WEBER A. R.; PROTE J.: Collaboration Mechanisms to increase Productivity in the Context of Industrie 4.0. *ScienceDirect* [online]. 2014 [cit. 2017-03-06]. Dostupné z: <http://www.sciencedirect.com/>.

[10] INDUSTRY 4.0 FOR BEGINNERS. *PLCdesign.xyz* [online]. 3.10.2016 [cit. 2017-03-06]. Dostupné z: <http://plcdesign.xyz/en/industry-4-0-for-beginners/>.

[11] ZEZULKA, František a Ondřej HYNČICA.: Průmyslový Ethernet II: Referenční model ISO/OSI. *Automa: časopis pro automatizační techniku*. 2007 [cit. 2017-03-07]. Praha: FCC Public, 2007, č. 3.

[12] ZEZULKA, František a Ondřej HYNČICA.: Průmyslový Ethernet VII: Přehled současných standardů. *Automa: časopis pro automatizační techniku*. 2008 [cit. 2017-03-07]. Praha: FCC Public, 2008, č. 2.

[13] SÉRIOVÁ KOMUNIKACE PROGRAMOVATELNÝCH AUTOMATŮ TECOMAT - MODEL 32 BITŮ. *Tecomat: Programovatelné automaty*. TECO 2013, 184 stran.

[14] KOMINEK, Darek a P. Eng. ALBERTA.: OPC: The Ins and Outs to What It's About. *MatrikonOP*. 2009 [cit. 2017-03-08]. Kanada, 2009.

[15] FOXON. CO JE OPC? OPC SERVER? OPC KLIENT? [online]. [cit. 2017-03-08]. Dostupné z: <https://www.foxon.cz/cs/blogs/80-co-je-opc-opc-server-opc-klient-.html>.

[16] KRITHI RAMAMRITHAM: Real-time Databases. *International Journal of Distribued nd Parallel Databases* [online]. Univesity of Massachusetts. 8.2.1996 [cit. 2017-03-10].

[17] F. Yang; E. Tschetter; X. Léauté; N. Ray; G. Merlino; D. Ganguli: Druid:A Real-time Analytical Data Store. *Matemarkets* [online]. USA. 2014 [cit. 2017-03-10].

[18] SIEMENS. Programming an OPC UA .NET Client with C# for the SIMATIC NET OPC UA Server [online]. [cit. 2017-04-15]. Dostupné z: <https://support.industry.siemens.com/cs/document/42014088>.

[19] SIEMENS: Connecting a PC Station to an S7-1200 using OPC. SIEMENS 2015, 71 stran.

[20] MELSOFT: MX OPC Server Quick Start. *MX OPC Server Quick Start*. MELSOFT 2014, 47 stran.

[21] TECO: Komunikační PLCCOmS. *Uživatelská příručka*. TECO 2017, 27 stran.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AES	Advanced Encryption Standard
CD	Kompaktní disk – Compact Disc
ECC	Elliptic Curve Cryptography
EPC	Elektronický kód produktu – Electronic Product Code
GSM	Globální systém pro mobilní komunikaci – Global System for Mobile Communication
ID	Identifikace
IEEE	Institut pro elektrotechnické a elektronické inženýrství – Institute of Electrical and Electronics Engineers
IIoT	Průmyslový Internet věcí – Industrial Internet of Things
IoT	Internet věcí – Internet of Things
IP	Internet Protocol
LTE	Long Term Evolution
mbar	Milibar - jednotka tlaku
mbar.l/s	Jednotka netěsnosti/úniku (za normálních pokojových podmínek dochází ke ztrátě 1 krychlového centimetru plynu za sekundu)
ms	Milisekunda - jednotka času
NFC	Komunikace na krátkou vzdálenost – Near Field Communication
OLE	Object Linking and Embedding
OPC	OLE for Process Control
PLC	Programovatelný logický automat – Programmable Logic Controller
PKI	Infrastruktura veřeného klíče – Public Key Infrastructure
RFC	Žádost o komentáře – Request For Comments
RFID	Identifikace na rádiové frekvenci – Radio Frequency Identificator
SSH	Secure Shell

TCP/IP	Transmission Control Protocol/Internet Protocol
uCode	Všudypřítomný kód – Ubiquitous Code
UDP	User Datagram Protocol
UMTS	Univerzální mobilní telekomunikační systém – Universal Mobile Telecommunication System
UWB	Ultra-širokopásmové – Ultra Wide Band
VPN	Virtuální soukromá síť – Virtual private network
WiFi	Bezdrátová věrnost – Wireless Fidelity
WPAN	Bezdrátová osobní síť – Wireless Personal Area Network

SEZNAM PŘÍLOH

A OBSAH PŘILOŽENÉHO CD

Obsah přiloženého CD je uspořádán následujícím způsobem:

- `Hlavní dokument` ..... Dokument diplomové práce v pdf
  - `diplomova-prace-pavel-marek.pdf`
- `Přílohy` ..... Přílohy diplomové práce
  - `Zdrojové kódy aplikace` ..... Zdrojové kódy desktopové aplikace
    - `Contol-Viewing-Software`
  - `Zdrojové kódy PLC a OPC` ..... Zdrojové kódy PLC a OPC Siemens
    - `S7-1214C`
    - `S7-1215C`
    - `OPC_Many-PLCs-Config-V2`
  - `Konfigurační soubory` ..... Soubory pro konfiguraci OPC serverů
    - `Simatic NET V12`
      - `Many-PLCs-Config.xdb`
    - `MX OPC SEVER`
      - `MXConfiguration.mdb`
    - `PLCComS`
      - `PLCComS.ini`
  - `Manuály` ..... Manuály pro konfiguraci OPC serverů
    - `Siemens-OPC.pdf`
    - `Mitsubishi-OPC.pdf`
    - `Teco-OPC.pdf`
  - `Aplikace pro chytrou továrnu` ..... Spustitelný soubor aplikace
    - `Release`
  - `Databáze` ..... Soubor databáze a nástroj pro prohlížení
    - `pmbase.db`
    - `SQLiteStudio`