



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

REALIZACE DIGITÁLNÍHO DVOJČETE S POMOCÍ AZURE DIGITAL TWIN

IMPLEMENTING A DIGITAL TWIN WITH AZURE DIGITAL TWIN

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tomáš Dvořák

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Václav Kaczmarczyk, Ph.D.

BRNO 2025



Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Tomáš Dvořák

ID: 220974

Ročník: 2

Akademický rok: 2024/25

NÁZEV TÉMATU:

Realizace digitálního dvojčete s pomocí Azure Digital Twin

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s principem digitálního dvojčete a technologií Azure Digital Twin.
2. Realizujte propojení vybraného fyzického zařízení (technologického celku) prostřednictvím Azure IoT Hub, sběr dostupných informací a přenos k digitálnímu dvojčeti.
3. Realizujte digitální dvojče - model zařízení, s pomocí kterého bude možné provádět virtuální experimenty .
4. Prostřednictvím platformy MS Azure propojte fyzické zařízení a digitální dvojče.

DOPORUČENÁ LITERATURA:

S. V. Nath, P. Schalkwyk: Building Industrial Digital Twins, Packt Publishing 2021. ISBN: 978-1-83921-907-8

Dle pokynů vedoucího práce.

Termín zadání: 10.2.2025

Termín odevzdání: 21.5.2025

Vedoucí práce: doc. Ing. Václav Kaczmarczyk, Ph.D.

doc. Ing. Petr Fiedler, Ph.D.

předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Digitalizace výrobních procesů je v Průmyslu 4.0 klíčová pro zvýšení efektivity. Jedním z představitelů této transformace je digitální dvojče. Diplomová práce se zabývá implementací digitální dvojčete v cloudové platformě Microsoft Azure. Práce využívá služby Microsoft Azure a běžně využívané komunikační principy a protokoly pro spojení fyzického a virtuálního světa. Výsledkem je digitální dvojče, které umožňuje vizualizaci nebo zpětné ovládání technologického procesu. Práce tak nabízí návod, jak lze s využitím cloudových technologií vytvořit digitální dvojče zařízení.

KLÍČOVÁ SLOVA

Průmysl 4.0, Digitální dvojče, Internet věcí, Microsoft Azure, IoT Hub, MQTT, Modbus TCP, Cloud

ABSTRACT

The digitalization of manufacturing processes is a key factor in increasing efficiency within Industry 4.0. One of the main representatives of this transformation is the digital twin. Master thesis focuses on the implementation of a digital twin using the Microsoft Azure cloud platform. It utilizes various Azure services as well as commonly used communication principles and protocols to bridge the physical and virtual worlds. The result is a digital twin that enables visualization or remote control of the technological process. The work provides a practical guide on how to create a digital twin of a device using cloud technologies.

KEYWORDS

Industry 4.0, Digital Twin, Internet of Things, Microsoft Azure, IoT Hub, MQTT, Modbus TCP, Cloud

DVOŘÁK, Tomáš. *Realizace digitálního dvojčete s pomocí Azure Digital Twin*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2025. Vedoucí práce: doc. Ing. Václav Kaczmarczyk, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Tomáš Dvořák
VUT ID autora:	220974
Typ práce:	Diplomová práce
Akademický rok:	2024/25
Téma závěrečné práce:	Realizace digitálního dvojčete s pomocí Azure Digital Twin

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno

.....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

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

Obsah

Úvod	10
1 Průmysl 4.0	11
1.1 Princip Průmyslu 4.0	12
1.2 Kyberfyzikální systémy	13
1.3 Integrace Průmyslu 4.0	14
1.4 Pilíře Průmyslu 4.0	15
1.5 Digitální dvojče	18
1.5.1 Rozdělení	19
1.5.2 Způsoby návrhu	20
1.5.3 Aplikace	22
1.6 Prostředky komunikace	22
1.6.1 MQTT	22
1.6.2 Mosquitto	24
1.6.3 Modbus TCP	24
2 Microsoft Azure	26
2.1 Azure Digital Twins	27
2.2 Azure IoT Hub	28
2.3 Azure IoT Explorer	28
2.4 Azure Data Studio	29
3 Technologický proces	30
3.1 ISA-88	31
3.1.1 Procesní model	31
3.1.2 Fyzický model	32
3.1.3 Procedurální model	32
3.1.4 Řízení	33
3.2 Simulátor	33
3.2.1 TwinCAT	34
4 Realizace digitálního dvojčete	37
4.1 Integrace TwinCATu s Azure IoT Hubem	37
4.1.1 Vytvoření IoT Hubu	37
4.1.2 Autentizace klienta	39
4.1.3 Publikování dat	41
4.2 Digitální model v Microsoft Azure	43
4.2.1 Vytvoření Azure Digital Twins	43

4.2.2	Vytvoření modelu digitálního dvojčete	44
4.2.3	Propojení s 3D Scenes	48
4.3	Zpracování a přenos dat	50
4.3.1	Zpracování dat pomocí Azure Funkce	50
4.3.2	Databáze	53
4.3.3	Zpětná komunikace	54
4.4	Metody realizace	54
4.4.1	Využití obou metod	56
4.5	Fyzická vrstva	56
4.5.1	Propojení s fyzickým panelem	56
4.5.2	Komunikace přes Modbus	59
Závěr		60
Literatura		61
A Obsah elektronické přílohy		65

Seznam obrázků

1.1	Přechod k Průmyslu 4.0	15
1.2	Pilíře Průmyslu 4.0	17
1.3	Proces tvorby digitálního dvojčete	19
1.4	Rozdělení digitálního dvojčete	20
2.1	Domovská stránka služeb Microsoft Azure	26
2.2	Graf digitálních dvojčat	27
3.1	Propojení modelů podle ISA-88	32
3.2	Simulátor Pasterizační jednotky	34
3.3	Seznam použitých licencí	35
4.1	Datový tok z TwinCATu do Microsoft Azure	37
4.2	Přehled využití IoT Hubu	38
4.3	Generování SAS tokenu pomocí Azure CLI	40
4.4	Parametry pro připojení k IoT Hubu	41
4.5	Příchozí zprávy do IoT Hubu	42
4.6	Model Valve v jazyce DTDL	44
4.7	P&ID diagram Pasterizační jednotky	46
4.8	Řez hlavního tanku v Siemens NX	47
4.9	Editace widgetů v 3D Scenes	49
4.10	Digitální dvojče v 3D Scenes	50
4.11	Blokové schéma procesu	51
4.12	Volání Azure funkce	52
4.13	Ukázka naplněné databáze	53
4.14	Propojení procesu	55
4.15	Průmyslový panel Beckhoff	57
4.16	Nastavení síťových parametrů v počítači	58

Úvod

Diplomová práce se zabývá využitím služeb cloudové platformy Microsoft Azure. Microsoft Azure využívá globální dostupnosti internetu, bez něhož si dnešní svět dokážeme jen těžko představit. Internet se stal součástí nejen každodenního života, ale také výrobního průmyslu. Celý koncept Průmyslu 4.0 je na využití internetu a digitálních technologií založen. Cílem diplomové práce je propojit principy Průmyslu 4.0 a platformu Microsoft Azure v podobě vytvoření digitálního dvojčete s využitím jeho nabízených služeb.

Průmysl 4.0, neustále propíraný termín posledních let, cílí na zefektivnění výrobního procesu pro splnění poptávek zákazníků, jejichž požadavky jsou čím dál více specifické. Tyto požadavky mohou zavedené výrobní procesy naplnit bez zásadní transformace jen stěží. Průmysl 4.0 představuje automatizaci, digitalizaci a propojení všech částí výrobního řetězce. Digitalizací výroby vznikne obrovské množství dat, které je nutné někde uložit. Nejjednodušší a nejspolehlivější volbou jsou cloudová úložiště, která zajišťují oprávněným uživatelům přístup odkudkoli a kdykoli.

Jednou z klíčových technologií, která reprezentuje přechod na Průmysl 4.0, je digitální dvojče. Zatímco digitální model usnadňující návrh a simulaci chodu fyzického zařízení je ve výrobních procesech samozřejmostí, reálné digitální dvojče, které nasbíraná data zanalyzuje a na základě nich provede v reálném čase změnu fyzického zařízení, už nikoli.

První kapitola popisuje Průmysl 4.0, digitální dvojče a komunikaci, která je zásadní pro implementaci. Druhá kapitola představuje platformu Microsoft Azure a třetí kapitola se věnuje technologickému procesu, který tvoří základ pro digitální dvojče. Čtvrtá kapitola pak detailně popisuje jednotlivé kroky k vytvoření digitálního dvojčete. Zahrnuje využití služeb Microsoft Azure i implementaci zpětné komunikace s fyzickým zařízením.

1 Průmysl 4.0

V průmyslu se v dnešní době často skloňuje pojem Průmysl 4.0, který představuje čtvrtou průmyslovou revoluci. Té však předcházela dlouholetý vývoj spojený s různými technologickými pokroky. Po mnoho let lidé vykonávali různá řemesla se snahou vymýšlet a rozvíjet všechna průmyslová odvětví tehdejší doby. Klíčovým vynálezem pro první průmyslovou revoluci byl na konci 18. století tkalcovský stav a parní stroj, díky nimž se začalo přecházet z ruční výroby na strojní. Na to navazuje o století později objev elektrické energie, který byl charakteristický pro další průmyslovou revoluci. Začalo se s dělbou práce, načež byla vytvořena první montážní linka, která se zároveň stala elektrifikovanou. Rozvoj elektroniky a informačních technologií vedl v roce 1969 ke vzniku prvního PLC. Označuje se tím počátek třetí průmyslové revoluce, jelikož došlo k automatizaci procesů a jejich digitalizaci. Masové rozšíření internetu pak vedlo k další evoluci průmyslové výroby, konkrétně k propojování fyzických zařízení se softwarovými systémy. [1]

Průmysl 4.0 byl oficiálně představen na veletrhu v Hannoveru v roce 2013. Jeho cílem je vznik automatizované výroby bez přítomnosti člověka pomocí informačních a komunikačních technologií. Předpokladem pro vznik těchto chytrých továren jsou kyberfyzikální systémy, které jsou nezávislé, plně kompetentní a mezi sebou vzájemně propojené. Kyberfyzikální systémy spojují fyzický a digitální svět, kdy jsou díky senzorům a řídicí jednotce schopny sbírat veškerá data z výroby a komunikovat s ostatními kyberfyzikálními systémy.

Nasazení Průmyslu 4.0 do výroby s sebou přináší mnoho výhod. Umožňuje lépe reagovat na individuální požadavky zákazníků a zároveň pružně přizpůsobovat výrobu poptávce, což má za následek zvýšení produktivity výroby. Odstraněním prostojů ve výrobě se sníží náklady a digitalizací celé výroby dochází k udržení aktuálního stavu zásob nebo sledování dodávky výrobku až ke koncovému zákazníkovi. Díky menší spotřebě materiálů a energie se zároveň celá výroba stane více ekologičtější. Existuje mnoho dalších výhod, ale zjednodušeně řečeno dochází ke zvýšení efektivity výroby.

Vybudování chytré továrny dle principů Průmyslu 4.0 s sebou nese nemalé výzvy. Může to být velká počáteční investice, rekvalifikace zaměstnanců nebo zajištění bezpečnosti. Bezpečnost je v Průmyslu 4.0 zásadní, jelikož připojení na internet zvyšuje riziko kybernetických útoků. Dalším důležitým tématem je standardizace, kdy je klíčové najít vzájemnou řeč napříč celým výrobním řetězcem k zajištění spolupráce jednotlivých systémů. [2]

1.1 Princip Průmyslu 4.0

Příchodem Průmyslu 4.0 dochází ke změně architektury, kde klasickou automati-zační pyramidu s hierarchicky uspořádanými prvky nahrazuje decentralizované řízení. Decentralizované řízení je distribuované, flexibilní a inteligentnější. Tento přechod zapříčinily zejména stále specifitější a individuálnější požadavky zákazníků. Myšlenka Průmyslu 4.0 představuje zavedení vysokého stupně digitalizace, komunikaci přes internet věcí, využití principů umělé inteligence, zavedení bezpečnostních postupů a standardizaci veškerých kroků, struktur či protokolů.

Základem této architektury a digitální transformace podniku je pojem chytrá továrna. Je to základní stavební kámen Průmyslu 4.0, představující automatizovaný výrobní systém, kde fyzické a digitální prvky spolu komunikují v reálném čase. Chytrá továrna by měla dle konceptu Průmyslu 4.0 splňovat těchto šest základních principů:

- **Interoperabilita** - Smyslem interoperability je schopnost kyberfyzikálních systémů autonomně si vyměňovat data, zahajovat výrobní úkony a řídit se. Zahrnuje také schopnost vzájemně spolupracovat s ostatními systémy, jako je ERP (Enterprise Resource Planning) nebo MES (Manufacturing Execution Systems). Nejedná se tak pouze o stroje, ale o integraci všech softwarových a hardwarových systémů.
- **Virtualizace** - Umožňuje vytvoření digitálních kopií fyzických procesů a zařízení pro zrychlení návrhu, včasné detekování problémů a testování změn. Snahou je vytvořit digitální dvojče, které bude propojeno s fyzickým zařízením a v reálném čase optimalizovat výrobní proces.
- **Decentralizace** - Přenesení odpovědnosti za rozhodování na jednotlivé systémy a zařízení bez nutnosti čekat na pokyn z centrálního bodu. Zajišťuje rychlou reakci na změny a požadavky výroby.
- **Práce v reálném čase** - Základní požadavek Průmyslu 4.0 se zaměřuje na rozhodování v reálném čase. Tato schopnost kyberfyzikálních systémů vede k menší spotřebě zdrojů a energií. Je klíčová pro monitoring kvality a optimalizaci výrobních procesů.
- **Modularita** - Přizpůsobuje výrobní proces měnícím se požadavkům přidáváním nebo odebíráním jednotlivých modulů. Reaguje tak na změny v poptávce bez ztráty nákladů a produktivity. Je také důležitá pro implementaci nových technologií jako reakce na změny trhu.
- **Orientace na služby** - Jedná se o zpřístupnění lidských, obchodních a kyberfyzikálních služeb prostřednictvím internetu. Umožňuje spolupráci nejen mezi jednotlivými závody v rámci jedné společnosti, ale také s partnerskými společnostmi a zákazníky. [3]

1.2 Kyberfyzikální systémy

Kyberfyzikální systémy (CPS) představují propojení fyzického světa s digitálním. Vycházejí z mechatronických systémů, kdy zahrnují integraci senzorů a akčních členů s výpočetními a softwarovými systémy k zajištění vzájemné komunikace. Celý podnik se zpravidla skládá z několika CPS, které jsou schopny mezi sebou komunikovat. Komunikace mezi jednotlivými CPS probíhá v reálném čase na základě IoT. Klíčovým rysem těchto systémů je schopnost samostatného rozhodování a učení na základě modelu prostředí a komunikace s ostatními CPS.

V průmyslové výrobě se pro CPS používá pojem kyberfyzikální výrobní systémy (CPPS). Jedná se o integrované celky složené z výrobních zařízení, lidí, produktů a softwarových systémů. Na základě komunikačních rozhraní společně monitorují a realizují výrobní operace. CPPS umožňují nejen sběr a vyhodnocování dat v reálném čase, ale i jejich sdílení a využití pro optimalizaci výroby, prediktivní údržbu či autonomní řízení výroby. Ve spolupráci s ostatními CPPS vytvářejí komplexní výrobní linku, která je představitelem konceptu chytré továrny. [4]

Pro implementaci CPS krok za krokem byla navržena architektura, která je rozdělena do pěti úrovní. Tzv. 5C architektura je zaměřena primárně na vertikální integraci. Představuje jednotlivé úrovně od získávání dat až po zpětnou vazbu a autonomní řízení.

- **Konektivita** - Úroveň zahrnující sběr dat ze senzorů, strojů a řídicích systému ERP a MES. Důležitý je výběr specifikace senzorů a přenosového protokolu pro předávání dat do úložiště.
- **Konverze** - Tato úroveň slouží k převodu získaných dat na užitečné informace. Jsou zde využity predikční algoritmy a nástroje pro monitorování stavu systému.
- **Kybernetika** - Třetí úroveň je prostředník mezi sběrem dat a rozhodovacími procesy. Shromažďuje totiž všechna data, čím vytváří informační síť pro jejich srovnání napříč podnikem. Analyzuje výkon jednotlivých strojů, vyhodnocuje historická data a provádí predikci životnosti.
- **Kognitivita** - V této úrovni se reprezentují relevantní informace uživateli a provádí se správné rozhodování. Díky hlubšímu porozumění systému lépe stanovuje priority údržby. Zároveň podporuje vzdálenou diagnostiku a spolupráci více odborníků.
- **Konfigurace** - Poslední úroveň slouží jako zpětná vazba mezi kybernetickým a fyzickým prostředím. Umožňuje strojům samostatně se přizpůsobovat a nastavovat. [5]

CPS nacházejí uplatnění nejen ve výrobním průmyslu, ale i v dopravě, energetice nebo zdravotnictví. V oblasti energetiky mohou autonomní CPS řídit způsob využití vyrobené elektrické energie na základě její známé ceny. Ve zdravotnictví se řeší otázka kardiologických operací bez nutnosti, byť na malou chvíli, zastavit srdce. Možnou aplikací je CPS, který by neustále srdce monitoroval a porovnával s detailním modelem. Propojoval by tak robotiku, kamerový systém a model srdce. Bylo by možné, aby robot provedl přesný zákrok, ačkoliv by srdce bylo stále v pohybu. Celý systém však musí být extrémně přesný, bezpečný a robustní.

I ze zmíněných praktických příkladů vyplývá, že podstatou CPS je spojení fyzického a virtuálního světa. Fyzická zařízení mezi sebou komunikují prostřednictvím softwarových entit s cílem, aby ve virtuálním prostoru bylo možné průběh procesu nejen sledovat, ale i řídit. Pro tuto správnou funkci CPS je však zapotřebí mít přesný digitální model fyzického systému, tzv. digitální dvojče. Fyzický a virtuální prostor pak společně komunikují a spolupracují na základě řídicího algoritmu. [4]

1.3 Integrace Průmyslu 4.0

Průmysl 4.0 přináší nový princip propojení jednotlivých systémů. Mění se zavedený způsob komunikace od úrovně k úrovni a to v rámci vertikálního i horizontálního propojení.

Vertikální integrace

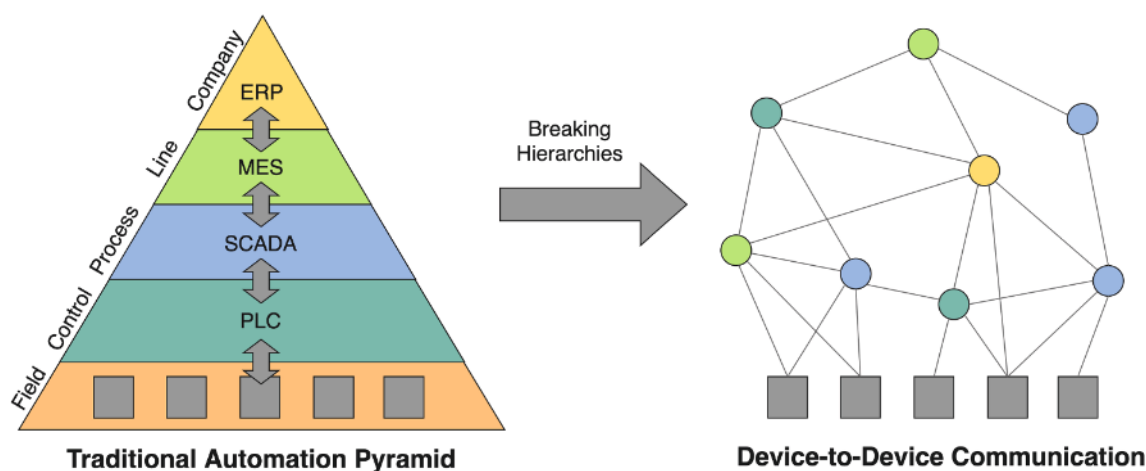
Před nástupem Průmyslu 4.0 byl základem pro řízení výroby MES systém ve spolupráci s ERP. Na MES systém navazoval monitorovací systém SCADA (Supervisory Control And Data Acquisition), který sbíral data z HMI panelů a PLC. V Průmyslu 4.0 již toto neplatí, přichází totiž decentralizace, neboli propojení každého fyzického stroje nebo softwaru s každým v celém výrobním řetězci.

Velkou roli zde hrají CPS, které umožňují výměnu dat mezi jednotlivými propojenými entitami. Samotný výrobek má v tomto konceptu svou inteligenci, zná postup, metody i jednotlivé kroky výroby, přičemž autonomně interaguje s CPS. V podniku pak musí existovat i entita, se kterou budou jednotlivé CPS vyjednávat, kdy se kromě výrobku jedná o pracovníky údržby. CPS spolupracují i s ostatními systémy, které data z výroby zpracovávají. Tradiční MES systém zde již neplní roli hlavního řídicího systému výroby, ale může být využit jako doplňková softwarová služba pro případy, kdy je potřeba centrální zásah nebo koordinace.

Horizontální integrace

Na rozdíl od vertikální integrace, která propojuje jednotlivé úrovně v rámci jednoho podniku, horizontální integrace její hranici přesahuje, kdy spojuje dodavatele, výrobce, koncové zákazníky i následný servis. K tomu je zapotřebí moderních informačních a komunikačních technologií, které umožní sdílení dat napříč celým řetězcem.

Cílem je vytvořit digitálně propojený systém, ve kterém všichni zmiňovaní účastníci vytvoří jeden kompaktní celek. Horizontální integrace výrazně pomáhá zkrátit reakční čas na výrobu, čímž se zefektivní výroba na individuální požadavky zákazníků. U příkladu automobilového průmyslu je podnik navázán na mnoho dodavatelů, kteří jsou taktéž napojeni na více výrobních podniků. Vše spočívá ve vzájemné interakci, kdy zákazník vytvoří objednávku vozu se specifickými požadavky, na což zareaguje podnik ve spolupráci s různými dodavateli. Prakticky v reálném čase poptá jednotlivé díly, jelikož v rámci softwarové platformy sdílejí svůj výrobní proces. Dodavatel zahájí ihned výrobu, případně dodá díl ze svého skladu, což výrazně sníží nutnost obrovských skladových zásob výrobního podniku. [4]



Obr. 1.1: Přejít k Průmyslu 4.0 [6]

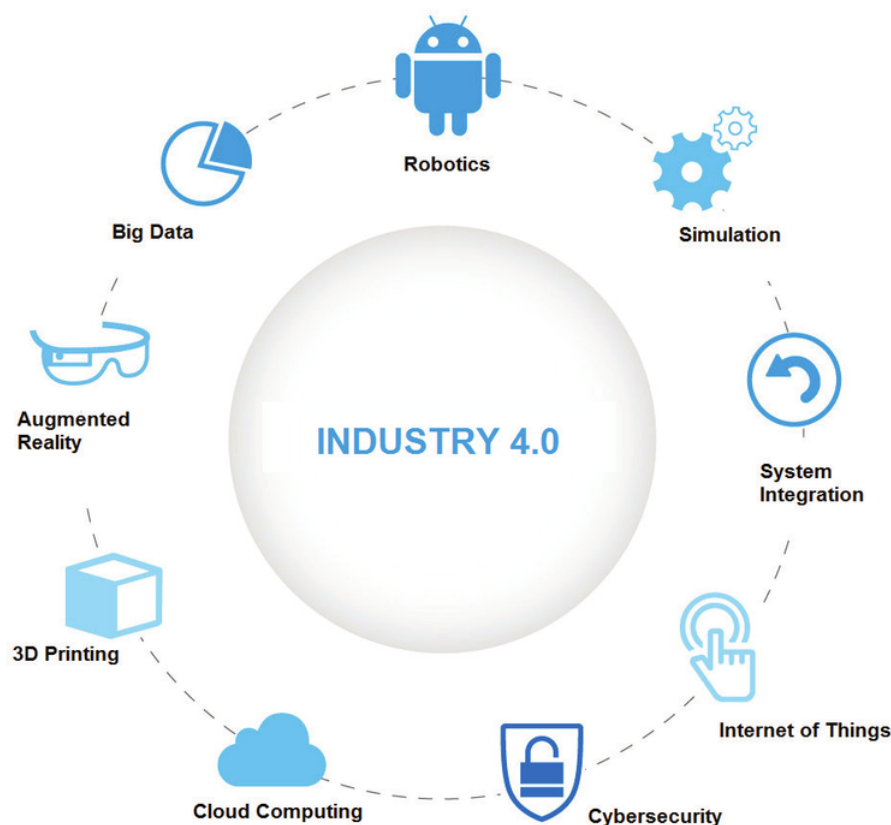
1.4 Pilíře Průmyslu 4.0

Transformace tradiční výroby na myšlenku Průmyslu 4.0 přináší 9 základních pilířů, které je potřeba do podniku implementovat:

- **Big data** - Řadí se sem všechna data ze senzorů, RFID, počítačů, GPS zařízení i sociálních sítí. Tato data by pak měla být kombinována s relační databází pro jejich přehlednost. Na základě toho pak může společnost lépe tato data

analyzovat a činit důležitá rozhodnutí. Dat vzniká v průmyslu čím dál více a jejich analýza může získat informace, o kterých podnik ani netušil.

- **Internet věcí** - Díky možnosti připojení na internet se mohou mezi sebou IoT (Internet of Things) zařízení propojit, čímž vytvoří síť fyzických zařízení, které si mezi sebou vyměňují data. Všechna data ze senzorů jsou odesílány do cloudu, kde dochází k jejich zpracování a provádění dalších akcí.
- **Cloudová úložiště** - Cloud je služba pro ukládání dat, provoz serverů a zajištění infrastruktury prostřednictvím internetu. Výhodou je, že společnost platí jen za to, co využije, a lze k němu přistupovat kdykoliv a odkudkoliv. Existují tři hlavní typy cloudových služeb. V případě IaaS (Infrastructure as a Service) poskytovatel nabízí infrastrukturu, kterou si může zákazník přizpůsobit podle svých potřeb k provozu vysoce výkonných úloh. Jedná se o výpočetní prostředky, servery, úložiště a síťové prvky. Zákazník ušetří na výdajích spojených s hardwarem a zaplatí pouze za to, jak dlouho daný prostředek využíval. SaaS (Software as a Service) nabízí hotový software, jako je Microsoft Office 365 nebo Google Apps. Poskytovatel v rámci smlouvy o poskytování služeb spravuje software i hardware a umožňuje využívat aplikaci s minimálními náklady. Výhodou je přístup odkudkoliv jen prostřednictvím internetu bez jakékoliv instalace softwaru. Zákazník je však omezen funkcemi, které poskytovatel nabízí a všechna data se nacházejí mimo společnost. Výhodou je jednoduchá správa, údržba a aktualizace. PaaS (Platform as a Service) je kombinací předchozích dvou modelů, kdy poskytovatel nabízí platformu pro celý životní cyklus aplikace, tedy od vývoje, přes testování až po nasazení a aktualizace. Odpadají veškeré starosti s nákupem a správou softwarových licencí nebo zkrácení času vývoje využitím předpřipravených funkcí. Všechny modely mají své využití v různých scénářích, přičemž IaaS je vhodné pro společnosti, které chtějí mít kontrolu nad infrastrukturou, SaaS pro rychlé využití nabízených služeb a PaaS pro vývojové týmy. [7]
- **Kybernetická bezpečnost** - Průmysl 4.0 je založen na datech a jejich bezpečnost je velice zásadní. Kybernetická bezpečnost se obecně zaměřuje důvěryhodnost, integritu a dostupnost dat. Hrozbami mohou různé typy malwaru, jako viry nebo trojské koně. Mohou pocházet z USB, emailů či pochybných webových stránek. Komunikace musí probíhat podle bezpečnostních postupů a norem s využitím šifrování, autentizace, autorizace nebo pravidelné aktualizace.
- **Systémová integrace** - Horizontální integrace je základem spolupráce firem a jejich procesů od dodavatele, přes výrobu až po zákazníka. Vertikální integrace se zaměřuje na jednu firmu, kde propojuje systémy k lepší spolupráci jednotlivých úrovní.



Obr. 1.2: Pilíře Průmyslu 4.0 [8]

- **Aditivní výroba** - Aditivní výroba umožňuje rychle splnit personalizované přání zákazníků prakticky za nulového odpadu. Je to technologie tisku jedné vrstvy po druhé na základě 3D modelu. Maximálně využívá materiál, čímž snižuje zdroje i náklady.
- **Simulace** - Kopie fyzického zařízení do digitálního prostředí umožňuje testovat zařízení před i v průběhu chodu. Typickým příkladem simulace je digitální dvojče, které dramaticky snižuje náklady.
- **Roboti** - Autonomní roboti provádějí opakující se činnosti efektivněji než člověk bez jakéhokoliv dohledu a interakce. Oproti tomu kolaborativní roboti jsou schopni s lidmi spolupracovat v reálném provozu bez ochranných překážek, čímž se sníží provozní náklady. Jsou vybaveny velkým množstvím senzorů pro detekci lidí k zajištění bezpečnosti práce.
- **Rozšířená realita** - K fyzickému systému poskytuje v reálném čase digitální informaci. Pomocí kamer a algoritmu je cílem získat více informací k provedení daného úkolu. Rozšířená realita usnadňuje interakci se zařízením, kdy snižuje množství chyb a zkracuje čas potřebný pro opravu. [2, 9]

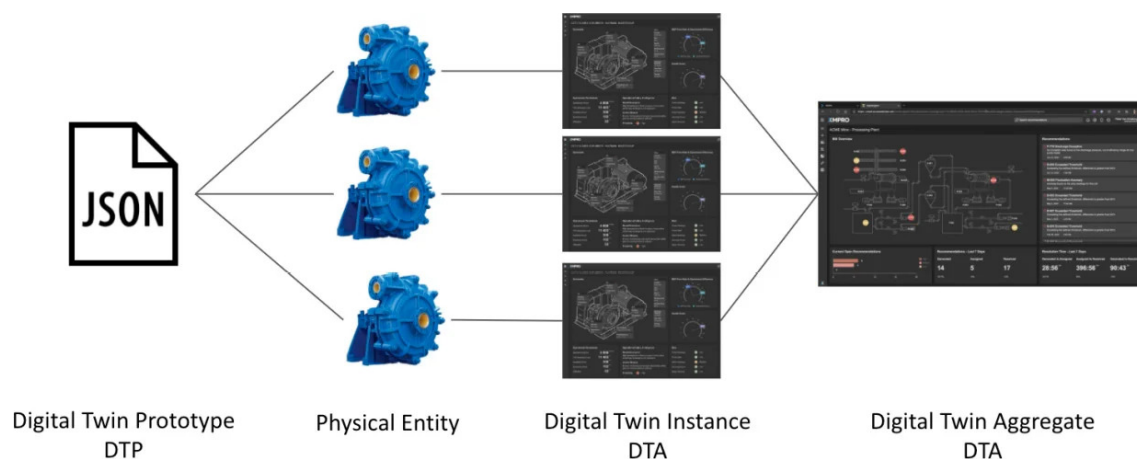
1.5 Digitální dvojče

Digitální dvojče je virtuální reprezentace fyzického systému, která v průběhu jeho celého životního cyklu monitoruje, analyzuje, predikuje a rozhoduje v reálném čase. Fyzický systém představuje jakýkoliv stroj, zařízení nebo celý proces. Mezi fyzickým systémem a digitálním dvojčetem probíhá neustálá výměna dat. Digitální dvojče obsahuje detailní informace o chování systému a za pomoci například nástrojů umělé inteligence je schopno data automaticky analyzovat, z čehož se získávají data ke zlepšení celkové efektivity.

Digitální dvojče je nástroj, který v posledních letech získal značnou pozornost a mnozí analytici se shodují, že má v průmyslovém prostředí obrovskou hodnotu. Termín digitální dvojče byl poprvé definován již v roce 2002 na univerzitě v Michiganu, kdy byl Johnem Vickers a Dr. Michaelem Grievesem prezentován v modelu Mirrored Spaces Model. Ti zjistili, že díky pokroku výpočetní techniky jsou systémy čím dál tím složitější, a tak byl potřeba nástroj, který poskytne lepší informace o fyzickém systému. Na základě toho definovali pojem digitální dvojče a charakterizovali jeho tři klíčové prvky:

- **Prototyp digitální dvojčete** - Představuje obecný model konkrétního typu fyzického zařízení. Nejedná se o výkres nebo 3D soubor zachycující fyzické rozměry, ale o informační model s popisem veškerých jeho vlastností a chování. Často je vytvořen v souborových strukturách, jako je JSON (JavaScript Object Notation) nebo XML (eXtensible Markup Language). Například pro dvacet stejných čerpadel existuje pouze jeden prototyp pro jeho popis, jelikož všechny mají stejné vlastnosti.
- **Instance digitální dvojčete** - Je to konkrétní instance každého fyzického zařízení vycházející z daného prototypu. Na základě jednoho prototypu může existovat pouze jedna instance nebo naopak mnohem více. Výrobní proces nelze popsat jedním prototypem, jelikož se skládá z několika zařízení, které mají vždy svůj prototyp.
- **Agregát digitálního dvojčete** - Kombinuje vytvořené instance do jednoho celku a tím přináší komplexní pohled na chování celého procesu. Díky němu lze například porovnávat výkon nebo provozní stav čerpadel, čehož by se takto efektivně na úrovni instancí nedosáhlo. [10]

Využití digitálního dvojčete se liší v závislosti na velikosti podniku a složitosti výrobního procesu. V menších podnicích často nenabízí nasazení digitálního dvojčete takové přínosy, jako tomu je v podnicích s rozsáhlými výrobními procesy. V těchto podnicích vzniká velké množství výrobních dat, které nelze identifikovat bez robustní IoT infrastruktury, tudíž nelze implementovat digitální dvojče k jejich vyhodnocení.



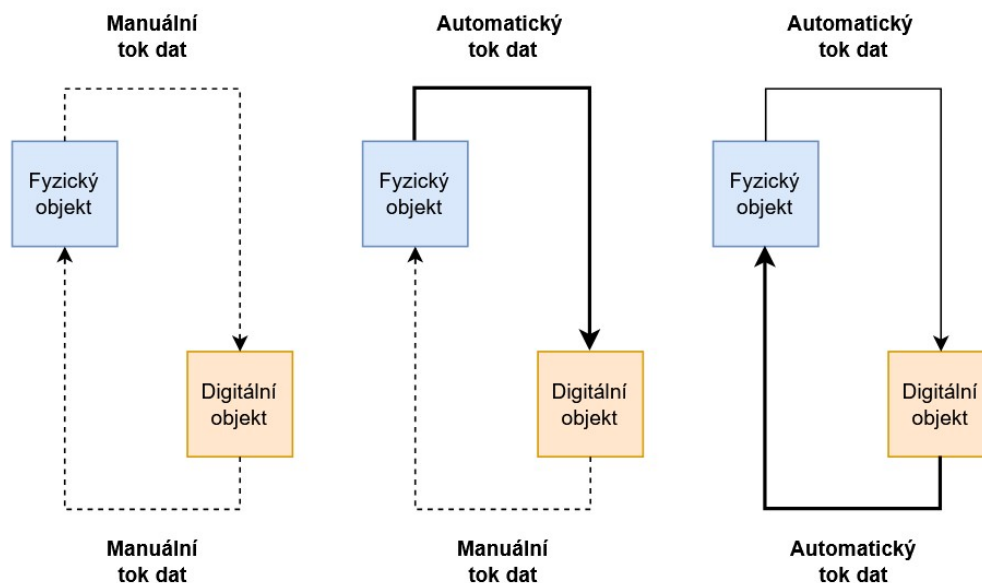
Obr. 1.3: Proces tvorby digitálního dvojčete [10]

Digitální dvojčata jsou dostupná odkudkoli a podporují spolupráci více týmů. To umožňuje efektivní vytváření modelu, testování v simulovaném prostředí a včasné odhalení problémů nebo špatné komunikace ještě před samotnou implementací. Tím dochází ke zrychlení vývoje i vyšší kvalitě. Digitální dvojče podrobně monitoruje fyzický systém a díky analýzám predikuje nejen budoucí chování, ale také nutnost údržby. Identifikace včasné údržby předchází neočekávanému zastavení výroby, pomáhá kvalitě výrobku kontrolou dodržování bezpečnostních norem a celkově k prodloužení délky a efektivity systému. [11]

1.5.1 Rozdělení

S digitálním dvojčetem se pojí termíny digitální model a digitální stín, které bývají, mnohdy i záměrně, nahrazovány pojmem digitální dvojče. Mezi těmito pojmy však existuje zásadní rozdíl spočívající ve výměně dat mezi fyzickým systémem a jeho digitální replikou.

Digitální model je vytvořen v první fázi, kdy se jedná o ručně vytvořenou digitální kopii již existujícího nebo plánovaného fyzického systému. Je to 3D model zahrnující fyzikální vlastnosti daného fyzického systému, ale neprovádí s ním žádnou interakci. Nedochází tedy k žádnému automatizovanému toku dat, tudíž nedochází k žádné automatické změně fyzického systému. Ta může nastat pouze manuálním zásahem. Digitální model je po celou dobu neměnný. Pokud však dojde k jakékoli úpravě fyzického systému, je zapotřebí digitální model aktualizovat. V rámci modelu však není nutné vytvářet doslova kopii fyzického systému, ale musí obsahovat ty aspekty, které jsou zásadní pro potřeby simulace.



Obr. 1.4: Rozdělení digitálního dvojčete [12]

Digitální stín reprezentuje aktuální stav fyzického systému. Do digitální repliky proudí data, která jeho stav neustále aktualizují. Jedná se o aktuální data ze snímačů, řídicích členů, ručních zásahů, ale také o všechna historická data, včetně opotřebení jednotlivých dílů. Každá změna se okamžitě projeví i v digitální replice, ale tato komunikace je pouze jednosměrná. Změnou fyzického systému dochází ke změně parametrů digitálního stínu. Digitální stín se díky neustálému přísunu dat používá zejména pro monitoring.

Teprve zajištění obousměrné automatické komunikace vzniká digitální dvojče. Existuje tak možnost na základě dat a řídicího algoritmu měnit stav fyzického systému. Tato technologie je výpočetně velice náročná, a proto skutečných digitálních dvojčat existuje omezené množství. [13, 14]

1.5.2 Způsoby návrhu

Tvorba digitálního dvojčete by se obecně dala rozdělit na dva způsoby. První způsob spočívá v použití digitálního dvojčete pro efektivní realizaci nových výrobních procesů a druhý pro jeho využití ke zlepšení a optimalizaci aktuálních procesů. V průmyslové výrobě již spousta procesů existuje, tudíž využití digitálního dvojčete pro vznik nových procesů je velmi omezené. V praxi se oba způsoby kombinují a doplňují, přičemž závisí na daném řešení. Každý z nich se skládá z několika specifických fází.

Nové výrobní procesy

První fáze zahrnuje vytvoření digitálního modelu fyzického systému před fyzickou realizací. Tento model je vytvářen v softwaru umožňujícím návrh i simulaci. Simulace pomáhá testovat systém za různých podmínek teplot, tlaků a zatížení. Testování probíhá až na hranici selhání pro vhodný výběr fyzických komponent. Tento způsob výrazně snižuje chyby, čas a náklady na výrobu nebo pozdější fyzické testování.

Po fyzické realizaci systému je digitální model přizpůsoben na základě specifikací systému a provozních podmínek. Provádí se další simulace chování pro testování různých změn vstupních parametrů a případných chyb. Digitální model je v této fázi propojen s fyzickým systémem a data z fyzického systému jsou ukládána na cloud pro jejich další zpracování.

Ve třetí fázi je fyzický systém uveden do provozu a z digitálního modelu se stává reálné digitální dvojče. Na základě analýzy dat z cloudu je digitální dvojče využito pro predikci chování, kdy je schopno v reálném čase rozhodovat o aktualizaci parametrů fyzického systému. Tato klíčová vlastnost automaticky optimalizuje proces, čímž zefektivňuje fyzický systém, jelikož nenastává žádný prostoj.

Zavedené výrobní procesy

V přípravné fázi je klíčové provést analýzu zaměřenou na kritická místa výrobního procesu. Odborníci z oblasti řízení zde odhalují podmnožinu zařízení, která způsobují většinu poruch a prostojů. Tyto poznatky pak identifikují ta zařízení, na která se zaměří vývoj digitálního dvojčete.

Prostřednictvím senzorů, IoT, ERP nebo MES systémů lze podrobně tato zařízení monitorovat, což zvyšuje hodnotu digitálního dvojčete. V první fázi IT týmy ověřují dostupnost těchto datových zdrojů a posuzují jejich kvalitu. Dále se definují klíčové faktory úspěchu, tedy konkrétní přínosy digitálního dvojčete v porovnání s náklady na zavedení relevantních datových zdrojů.

V další fázi nastává vývoj digitálního dvojčete, kdy se integruje na návrhové, výrobní a údržbové systémy. Ověřuje se správnost návrhu pomocí simulací a probíhá zkušební provoz. Zároveň zde začíná sledování metrik k vyhodnocení přínosů a možností škálovatelnosti řešení.

Ve třetí fázi se ověřuje, zda digitální dvojče splnilo stanovené přínosy na základě sledování dat v reálném čase a zpětné vazby od uživatelů. Dochází k doladění systému podle provozních zkušeností a posuzuje se jeho udržitelnost. Posledním bodem je pak rozhodnutí, zda se digitální dvojče naplno nasadí do produkce, nebo nikoliv.

V poslední fázi dochází ke škálování digitálního dvojčete na další procesy podniku. Přináší to dodatečné požadavky na funkcionalitu celého systému, ale díky tomu se z digitálního dvojčete stává nástroj pro digitální transformaci podniku. [10, 11]

1.5.3 Aplikace

Jak již bylo zmíněno, digitální dvojče se využívá ve výrobním průmyslu k simulaci a predikci každého kroku fyzického systému, pro návrh designu, prediktivní údržbu, snížení nákladů a celkově pro zlepšení efektivity výroby. Nachází tak uplatnění v celém životním cyklu zařízení. Využívá se i v jiných průmyslových odvětvích.

Ve zdravotnictví se používá digitální model lidského srdce pro pochopení srdečních onemocnění a jejich predikci. Aplikace digitálního dvojčete ve zdravotnictví je velice náročná, jelikož zde hraje velkou roli morální aspekt. V případě chyby to může stát člověka život, oproti pouhému prostoji ve výrobním průmyslu.

V zemědělství jsou velice složité a dynamické procesy z důvodu počasí, půdních podmínek nebo sezónnosti. Využití digitálního dvojčete se zde zatím neprosadilo tak, jak v ostatních odvětvích. Aplikace digitálního modelu zde může na základě předpovědi počasí a změn klimatu predikovat úrodu nebo při analýze půdy predikovat požadované množství hnojiva či vody. V ropném a plynárenském průmyslu pomáhá v analýze parametrů konstrukcí pro predikci stavů konstrukcí.

Ve stavebnictví již dlouho existují digitální modely budov. Obsahují veškeré informace o budově a slouží k její analýze, kdy dokážou simulovat jednotlivé kroky stavebního procesu a predikovat údržbu. Co se týče stavebnictví, tak nejznámějším digitálním modelem je město Singapur, kde je snaha o zlepšení infrastruktury nebo dopravy města. Po celém městě se nacházejí senzory, které sbírají data pro přesnější simulace. [14]

1.6 Prostředky komunikace

Efektivní a spolehlivá komunikace mezi fyzickými zařízeními a digitálním prostorem je naprosto zásadní. Pro přenos dat mezi senzory, řídicími jednotkami, servery a cloudovými službami se využívají různé komunikační protokoly. V této kapitole je popsán protokol MQTT s příkladem MQTT brokera a Modbus TCP.

1.6.1 MQTT

MQTT (Message Queuing Telemetry Transport) je otevřený komunikační protokol určený pro spolehlivý přenos dat, který umožňuje posílání krátkých a kompaktních zpráv, což zajišťuje flexibilní komunikaci. Byl vynalezen společností IBM a v roce 2013 byl předán organizaci OASIS. Ta následně provedla standardizaci a nadále ho rozvíjí. MQTT je snadno implementovatelný, jelikož zabírá minimum paměti i výpočetních prostředků. Díky těmto vlastnostem je ideální volbou pro komunikaci v rámci IoT.

Výměna zpráv mezi klienty probíhá prostřednictvím centrálního serveru, tzv. brokera. Celý proces komunikace funguje na principu publisher-subscriber, kdy publisher pošle zprávu s určitým tématem na server a subscriber tuto zprávu na základě tématu přečte. Broker zajišťuje distribuci zpráv mezi zařízeními a uchovává je, dokud je subscriber nepřečte. Typickým příkladem publishera je snímač, který odesílá data na server, z něhož pak řídicí jednotka, reprezentující subscribenta, zprávu přečte a dále ji zpracovává. Výhodou je, že zařízení nemusí být aktivní v jeden čas, ale při výpadku brokera je komunikace nefunkční.

MQTT využívá jako transportní vrstvu protokol TCP/IP (Transmission Control Protocol over Internet Protocol). Při nastavení komunikace musí být zvolen konkrétní síťový port. Nejčastěji používaný port je 1883 pro nešifrovanou komunikaci, tedy nezabezpečenou komunikaci. Oproti tomu port 8883 je určen pro šifrovanou komunikaci využívající standard TLS/SSL (Transport Layer Security/Secure Sockets Layer).

Zásadní pro komunikaci přes MQTT je téma zprávy, které je zapsáno hierarchicky a odděleno lomítky. Téma může být jednoúrovňové nebo víceúrovňové podle zástupného znaku. Znakem + se nahradí jedna úroveň, například když chce subscriber znát teplotu všech tanků. Naopak znak # nahrazuje více úrovní a slouží pro získání dat ze všech snímačů, kdy se tento znak v zápise tématu použije až na jeho konci.

MQTT je lehký protokol, což znamená, že zpráva musí obsahovat pouze téma a obsah. Obsah zprávy není nijak definovaný, její formát může být binární řetězec, textová podoba nebo JSON. Zpráva však může obsahovat další informace zajišťující spolehlivost komunikace. Může to být nouzová zpráva v případě odpojení subscribenta, příznak k uložení zprávy brokerem pro všechny další subscribenty daného tématu nebo kvalita služby QoS (Quality of Service). MQTT protokol podporuje tři úrovně kvality služby:

- **QoS 0** - Zpráva je odeslána bez potvrzení a není zaručeno její doručení. Publisher nedostává žádnou odpověď, takže zpráva bude doručena maximálně jednou.
- **QoS 1** - Zpráva je doručena minimálně jednou, ale hrozí duplicita zpráv. Publisher dostává potvrzení o přijetí zprávy.
- **QoS 2** - Zpráva je doručena právě jednou a je tak vyloučena jakákoliv duplicita. Je to nejbezpečnější, ale zároveň nejpomalejší přístup. [15]

1.6.2 Mosquitto

Mosquitto je open-source nástroj broker, respektive server. Podporuje publikování a odebírání zpráv na základě protokolu MQTT. Je navržen tak, aby byl efektivní a úsporný na prostředky, což ho umožňuje použít jak pro jednoduchá zařízení, tak pro plnohodnotné servery. Mosquitto podporuje všechny verze protokolu MQTT i různé operační systémy. Umožňuje komunikaci podle standardu TLS/SSL, je škálovatelný a běžně využívá porty 1883 a 8883. Může být používán z příkazového řádku nebo aplikací, jako je například TwinCAT. [16]

1.6.3 Modbus TCP

Modbus TCP je otevřený komunikační protokol určený pro přenos dat mezi zařízeními využívajícími Ethernet a protokol TCP/IP. Modbus byl představen již v roce 1979 společností Modicon, ale díky své otevřenosti, široké podpoře u většiny výrobců PLC a snadné implementaci je stále nejrozšířenější protokol v průmyslové automatizaci. Modbus TCP vznikl rozšířením protokolu Modbus RTU (Remote Terminal Unit), který byl navržen pro sériovou komunikaci na krátké vzdálenosti s využitím rozhraní RS-485, RS-422 nebo RS-232. Modbus TCP je vhodný především pro síťovou komunikaci na větší vzdálenosti. Ve srovnání s Modbus RTU nabízí Modbus TCP možnost jej využívat v již existujících sítích Ethernetu, nabízí vyšší rychlost přenosu dat a možnost přenosu většího objemu dat. Naopak neobsahuje kontrolní součet CRC (Cyclic Redundancy Check) k detekci chyby, je dražší a vyžaduje více bezpečnostních opatření.

Komunikace funguje na principu master-slave, kdy vše iniciuje master, který posílá dotazy na slave zařízení a čeká na jejich odpověď. Může komunikovat s více zařízeními a očekávat od nich různý typ dat. Typickým příkladem může být komunikace mezi PLC a senzory či akčními členy. Modbus obsahuje čtyři typy registrů, které mají svou specifickou adresovou oblast. Holdovací registry uchovávají informace o procesních hodnotách a mohou být čteny i zapisovány. Oproti tomu vstupní registry jsou pouze pro čtení, kdy obsahují aktuální hodnoty ze senzorů. Diskrétní vstupy nesou informace o binárních stavech různých ventilů a cívky o binárních stavech výstupů, které lze i nastavovat.

Formát zprávy v Modbus TCP obsahuje 7bytovou hlavičku, která zahrnuje identifikaci požadavku, typ protokolu, délku zprávy a identifikátor zařízení. Následuje kód funkce, který specifikuje, zda se jedná o čtení nebo zápis do registrů. Posledními dvěma byty jsou adresa prvního registru a počet požadovaných registrů. Odpověď od zařízení má podobný formát jako dotaz, ale její obsah závisí na funkčním kódu. V případě chyby se nejvyšší bit funkčního kódu nastaví na jedničku, což signalizuje chybovou odpověď.

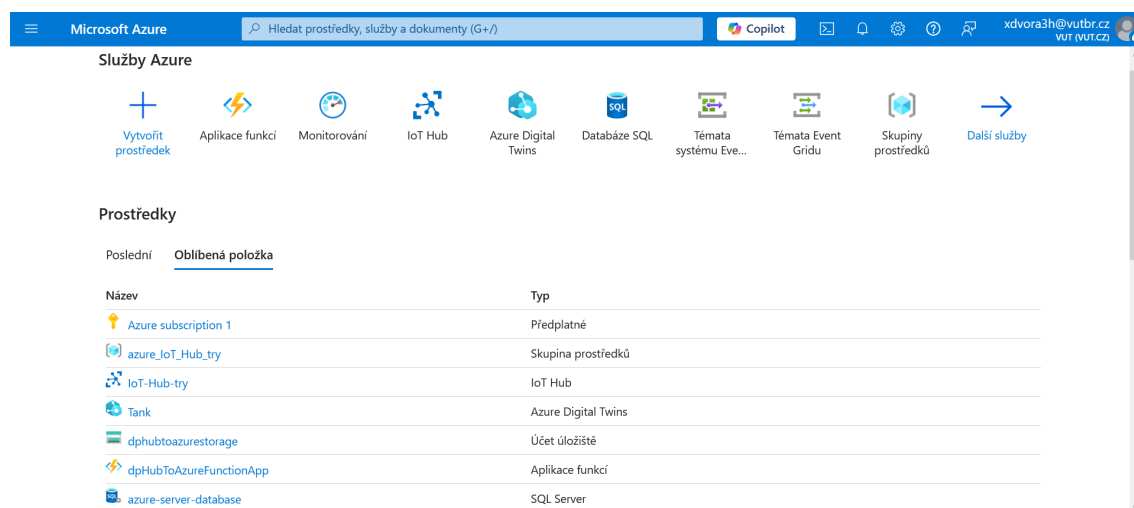
Zařízení v síti jsou identifikována IP adresou a komunikačním portem, přičemž standardní port pro Modbus TCP je 502. Zařízení musí být umístěna ve stejné podsíti. Modbus TCP obsahuje standardizované funkce pro snadnou diagnostiku a odstraňování poruch. Tím se výrazně zjednodušuje údržba a servis zařízení při komunikaci. Jak již bylo zmíněno, z hlediska bezpečnosti je Modbus TCP zranitelný, protože sám o sobě neobsahuje žádné integrované bezpečnostní funkce, jako je šifrování nebo autentizace. Proto se v průmyslovém prostředí různě kombinuje s dalšími protokoly a bezpečnostními mechanismy pro spolehlivý a bezpečný přenos dat. [17]

2 Microsoft Azure

Microsoft Azure je cloudová platforma společnosti Microsoft, která nabízí desítky služeb pro realizaci řady moderních aplikací. Služby, obsahující mnoho nástrojů a funkcí, jsou velice flexibilní, jelikož podporují veškeré operační systémy a programovací jazyky. Podporují integraci s různými vývojovými prostředími a nabízejí využití umělé inteligence, strojového učení nebo internetu věcí. Všechny služby jsou k dispozici online bez nutnosti instalace softwaru. Azure zahrnuje všechny tři typy cloudových služeb, kdy si zákazník může vybrat mezi správou kompletní infrastruktury, vývojem aplikací nebo využitím již vytvořených řešení. Podporuje také kombinaci svých datových center s jinými poskytovateli.

Microsoft Azure poskytuje bezpečnost, infrastrukturu, spolehlivost a škálovatelnost. Klíčovou vlastností Azure je zabezpečení dat a aplikací. Je to bezpečná platforma, která podporuje silnou autentizaci, řízení přístupu na základě rolí a šifrování dat. Pro správu prostředí Azure je možné využít Azure portál, příkazový řádek Azure CLI nebo PowerShell.

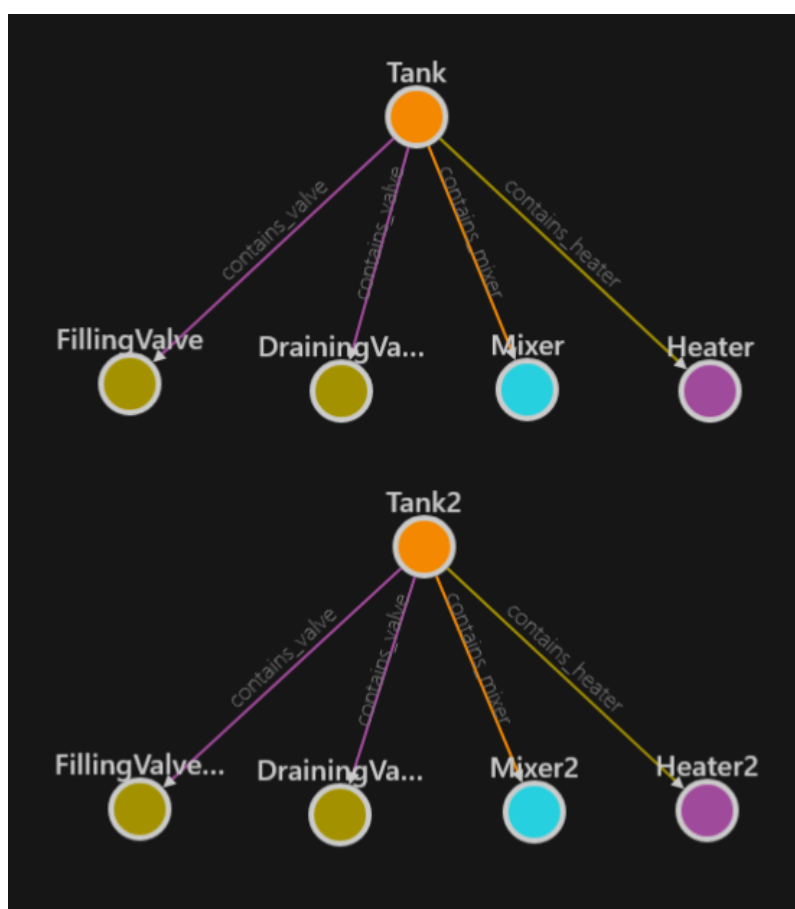
Pro využití služeb musí mít zákazník vytvořený účet a příslušné předplatné. Microsoft Azure nabízí režim průběžných plateb, kdy zákazník platí pouze za to, co skutečně využil. Pro případ testování existuje v tomto režimu určitý limit využití dat zdarma. Pro nové zákazníky platí měsíční bezplatné předplatné pro většinu služeb. [18]



Obr. 2.1: Domovská stránka služeb Microsoft Azure

2.1 Azure Digital Twins

Azure Digital Twins je služba typu PaaS, která nabízí digitální reprezentaci fyzického zařízení ve formě vytvoření vizuálního grafu digitálních dvojčat a relací mezi nimi. Jednotlivá fyzická zařízení mohou být připojena na IoT Hub a v reálném čase na základě přijatých dat aktualizovat svá digitální dvojčata v grafu. Tato služba slouží k získání přehledu o chování, k optimalizaci provozu a lepšímu rozhodování. Ve spolupráci s analytickými službami Microsoft Azure může vytvářet pokročilé aplikace.



Obr. 2.2: Graf digitálních dvojčat

Struktura Azure Digital Twins je složena z kruhů a čar, kdy kruhy reprezentují digitální dvojčata a čáry jejich vzájemný vztah. Nejprve se vytvoří model v jazyce DTDL (Digital Twin Definition Language), který obsahuje definici vlastností a vztahů. Digitální dvojče vznikne jako instance tohoto modelu, představující dané fyzické zařízení. Těchto instancí může z jednoho modelu vzniknout hned několik. Graf se zobrazuje v Průzkumníku, kdy je možné v horní liště pomocí SQL (Structured Query Language) dotazů sledovat jeho stav a zjišťovat vlastnosti. [19]

2.2 Azure IoT Hub

Azure IoT Hub je opět služba typu PaaS v cloudu, která slouží pro zabezpečenou obousměrnou komunikaci mezi připojenými zařízeními a aplikací IoT. Vytváří jakousi centrální bránu mezi zařízeními a cloudem, kdy spravuje odeslaná data. Komunikace probíhá prostřednictvím protokolů MQTT, AMQP (The Advanced Message Queuing Protocol) nebo HTTPS (Hypertext Transfer Protocol Secure), kdy je pro bezpečné připojení možné využít SAS tokeny (Shared Access Signatures) nebo certifikáty X.509. SAS token je řetězec generovaný službou Azure Storage ke zprostředkování přístupu k datům a službám Microsoft Azure. Obsahuje informace o tom, jaká služba má být zpřístupněna, jaké jsou umožněné operace a jak dlouho je SAS token platný. Certifikáty X.509, jako druhý bezpečnostní mechanismus, podporují standard TLS/SSL, kdy využívají k šifrování veřejný a soukromý klíč.

Je to služba typu back-end, kdy přijatá data uchovává pro další zpracování ostatních služeb. Připojit se k němu může prakticky jakékoli zařízení, přičemž Azure IoT Hub jednotlivá zařízení ověřuje, spravuje, sleduje a zajišťuje škálovatelnost. Podporuje nejen sběr a směrování telemetrických dat, ale také zasílání zpráv do zařízení nebo volání metod. Počet zpráv za den se mění v závislosti na úrovni účtu, v případě bezplatné verze zvládne 8000 zpráv za den na zařízení. To je vhodné pro malé aplikace nebo jednoduché testování, ale pro složitější aplikace nabízí Microsoft Azure vylepšení účtu s vyšší kapacitou zpráv. [20]

2.3 Azure IoT Explorer

Azure IoT Explorer je nástroj, který slouží pro správu všech zařízení připojených k IoT Hubu. Představuje alternativu k příkazovému řádku, jelikož se jedná o intuitivní grafické prostředí ke snadné orientaci mezi jednotlivými zařízeními. Nabízí seznam všech zařízení, informace o jejich stavu a jejich digitální reprezentaci. Připojením ke službě Azure Digital Twins se bude jejich stav na základě aktuálně přijatých dat aktualizovat. Umožňuje telemetrii, tedy sledování zpráv, které zařízení odesílají do IoT Hubu, a podporuje také odeslání zpráv do zařízení, včetně volání metod k jednoduššímu testování.

Azure IoT Explorer umožňuje se najednou připojit k několika IoT Hubům, což se využije v případě správy zařízení v různých projektech. Připojení k jednotlivým IoT Hubům probíhá pomocí připojovacího řetězce, který obsahuje potřebné informace k autentizaci. Výhodou je možnost vytvoření a ovládání simulovaných zařízení pro testování bez nutnosti fyzického hardwaru. Aplikace je ve stádiu Preview, kdy se neustále opravují nedostatky nebo přidávají nové funkce, tudíž není určena pro produkční řešení. [21]

2.4 Azure Data Studio

Azure Data Studio je nástroj od společnosti Microsoft určený pro správu a práci s databázemi. Umožňuje připojení k různým databázím, například k MySQL, SQL serveru nebo Azure SQL Database. Nabízí jednoduché a moderní vývojové prostředí pro práci s daty. Podporuje jazyk SQL ke spouštění dotazů a uložení výsledků ve formátu XML, CSV, JSON nebo Excel.

Azure Data Studio je vhodné nejen pro správu databází, ale také pro jejich analýzu a monitorování. Poskytuje nástroje jako integrovaný terminál, integrovaný Git, možnost rozšíření a podporu dotazů přes více serverů. Umožňuje spravovat databáze nasazené v prostředí cloudových služeb v testovacím i produkčním nasazení. Aplikace je dostupná zdarma a funguje na operačních systémech Windows, macOS i Linux. Azure Data Studio však bude oficiálně ukončeno 28. února 2026. Microsoft doporučuje přejít na Visual Studio Code, které nabízí více funkcí s širší podporou rozšíření a robustnějším vývojovým prostředím. [22]

3 Technologický proces

Pro realizaci diplomové práce byl vybrán projekt Pasterizační jednotky, jako komplexního technologického procesu, který byl využíván v rámci předmětu Automatizace procesů. Tento projekt splňuje požadavky standardu ISA-88 pro řízení dávkových procesů. Následující kapitola představující obecný popis pro pochopení jeho funkce vychází z publikací [23, 24].

Technologický proces se skládá z posloupnosti mechanických, chemických a fyzikálních operací, společně s následným skladováním, balením či přepravou, které mají za cíl přeměnit vstupní materiál na požadovaný výrobek. Tyto procesy lze rozdělit podle přístupu k výrobku na základě množství, výrobního času nebo specifikace zákazníka.

Spojité procesy

Spojité procesy jsou charakteristické tím, že výroba probíhá permanentně a výsledkem je souvislý tok vyráběného produktu. Využití spojitých procesů je zejména v potravinářském a energetickém průmyslu, kde je výhodou stabilní výroba daného produktu, jako je například elektřina.

Celá výroba je uzpůsobena výrobě jednoho typu produktu a existuje tak pouze malá možnost flexibility, kdy bývá přechod na jiný druh ekonomicky velice náročný. Z toho důvodu se produkty vyrábějí předem podle odhadované poptávky a následně jsou skladovány k pozdější distribuci. Tento lineární způsob výroby na sklad vyžaduje výrobu produktů se stabilní poptávkou a bez potřeby kvalifikované obsluhy.

Diskrétní procesy

U diskrétních procesů je produkt vyráběn na základě požadavků zákazníka, což znamená, že výsledné množství produktu je ve srovnání se spojitými procesy mnohem menší. Typicky se jedná o jednotlivé kusy až malé série produktu. Diskrétní proces nabízí velkou flexibilitu ve výrobě, kdy dokáže reagovat na změnu výrobního procesu. Nejčastěji se využívá v elektrotechnice a strojírenství, typicky při výrobě aut.

Výrobní technologie obsahuje stroje, které nejsou určeny pouze pro jeden konkrétní úkon. K jejich seřízení na provádění dalších operací je zapotřebí kvalifikovaných operátorů. Celá výroba je nákladnější než sériová výroba a vyžaduje řízení z nadřazených systémů výroby, jako je například MES systém. Důležitým faktorem řízení je plánování samotné výroby, které musí brát v potaz faktory, jako je priorita zakázek, dostupnost zařízení a dostupnost operátorů.

Dávkové procesy

Dávkové procesy kombinují prvky diskrétní a spojité výroby. Výsledným produktem je dávka, po níž následuje dávka stejného nebo jiného produktu s podobnými vlastnostmi. Dávkové procesy se využívají v potravinářském nebo chemickém průmyslu, kde je zásadní správné složení dané dávky.

Výrobní proces umožňuje pružnou reakci na poptávku trhu díky schopnosti výroby jiného produktu na stejném zařízení. To spočívá ve změně parametrů a operací podle zvolené receptury. Důležitou součástí dávkových procesů je i čištění, které zajišťuje vysokou kvalitu jednotlivých dávek.

3.1 ISA-88

Řízení výrobních procesů vychází z klasické automatizační pyramidy. Centrální podnikový systém ERP se stará o obchodní stránku celého podniku a vysílá příkazy na výrobu do MES systému, což je řídicí systém, který efektivně plánuje a spouští výrobní operace. Tuto strukturu popisuje norma ISA-S95. Norma ISA-S95 je mezinárodní standard pro integraci podnikových a řídicích systémů a konkrétně popisuje platformu MOM (Manufacturing Operations Management). MOM systémy rozšířily MES systémy o definici komunikace a integrace s ostatními systémy.

Pro dávkové procesy byla vyvinuta norma ISA-88, která se zabývá nejnižší vrstvou pyramidy. K ovládání jednotlivých částí technologie využívá PLC a k jejich řízení pak nadřazený systém typu Batch. Smyslem je rozdělit proces na části, které jsou organizovány do modelů. Norma ISA-88 definuje tři modely pro pochopení jejího principu.

3.1.1 Procesní model

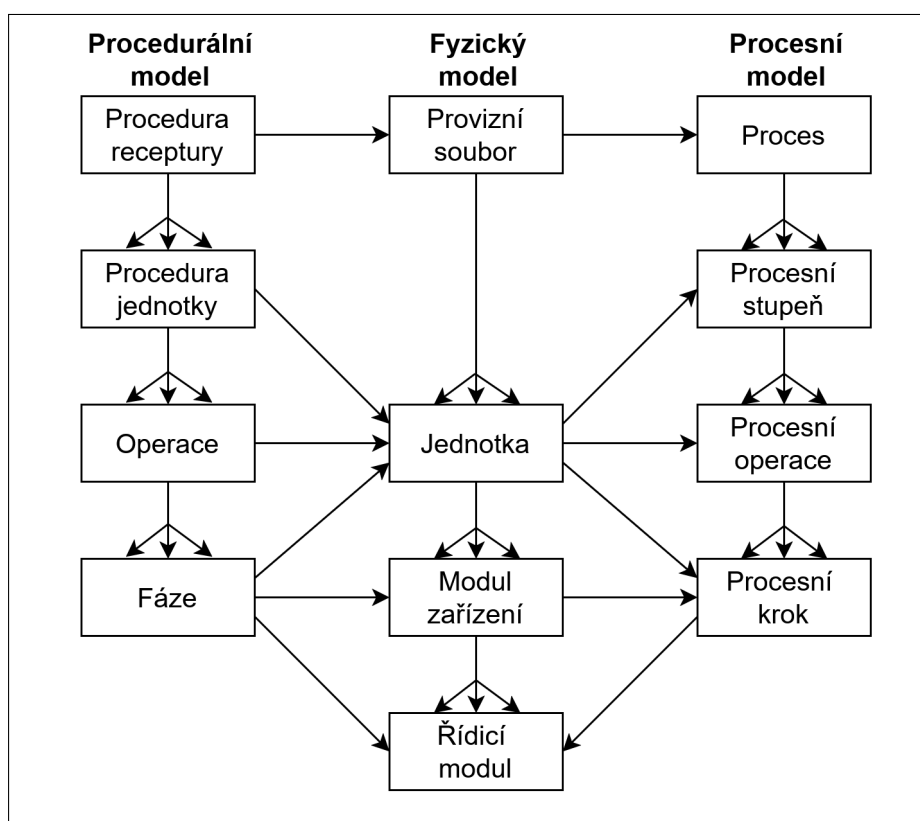
Definuje čtyřúrovňovou hierarchickou strukturu, která slouží k obecnému popisu výrobního procesu. Není jakkoliv vázán na fyzická zařízení, jelikož popisuje pouze postup výroby.

Proces, jako nejvyšší úroveň struktury, určuje, jaké činnosti se mají aplikovat na vstupní suroviny. Procesní stupeň se zaměřuje na jedinou část a nevyžaduje paralelní běh s ostatními stupni. Procesní operace představují hlavní procesní činnosti, jako je plnění nebo míchání. Procesní akce představuje konkrétní akci, například spuštění ohřevu nebo otevření ventilu.

3.1.2 Fyzický model

Popisuje fyzická zařízení potřebná k výrobě dané dávky produktu. Fyzický model vzniká společně s technologickými schématy, funkčními popisy a řídicími a elektrickými moduly.

Důležitou částí je procesní buňka, která zahrnuje všechny jednotky s cílem vytvořit finální produkt. Jednotka je hlavní část procesní buňky, která obsahuje všechna zařízení pro výrobu jedné dávky. Modul zařízení obsahuje více řídicích modulů s cílem přepravy dávky mezi nimi. Řídicí modul je nejnižší úroveň, která přímo ovládá akční členy, jako je motor nebo ventil.



Obr. 3.1: Propojení modelů podle ISA-88 [23]

3.1.3 Procedurální model

Spojuje předchozí dva modely k tomu, aby se daly popsat jednotlivé činnosti k výrobě jedné dávky. Tvoří základ pro softwarovou implementaci a skládá se ze čtyř úrovní.

V proceduře receptury je doménou procesní buňka, kdy se skládá z jednotkových procedur pro uskutečnění hlavní procesní akce. Procedura jednotky se skládá

z uspořádaného souboru operací, které vykonávají souvislou sekvenci, kdy se předpokládá pouze jedna aktivní operace v jednotce v jednom čase. Operace je množina fází k dosažení cílového produktu, kdy spouští a koordinuje jednotlivé fáze. Fáze realizuje konkrétní procesní akci.

3.1.4 Řízení

Řízení je rozděleno na tři typy. Základní řízení slouží pouze pro nastavování a udržování procesních veličin. Procedurální řízení vykonává jednotlivé fáze, kdy ovládá akce na zařízení tak, aby se odehrávaly ve správné sekvenci. Koordinační řízení pak na úrovni procesní buňky řídí postupy procedurálního řízení. Jeho snahou je koordinovat výrobu, aby všechny dávky byly včas hotové.

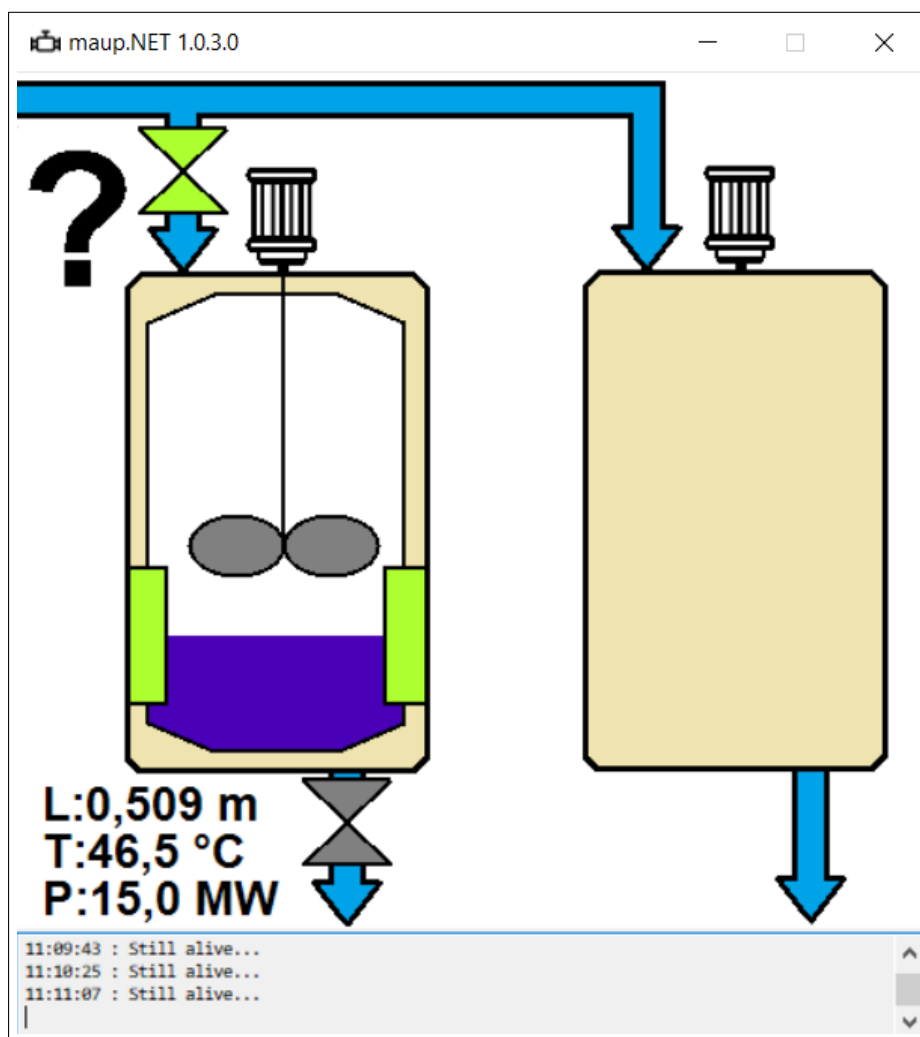
Výroba je řízena dle receptur, které obsahují parametry a operace pro daný typ produktu. Propojení s fyzickým procesem je na úrovni fáze, kde fáze posílá parametry do kódu PLC, který následně vrací reporty z fyzického procesu. Fáze se skládají z jednotlivých kroků a přechodů. V automatickém režimu se provádí vykonání procedury bez operátora, který může postup pouze pozastavit. Operátor nemůže ovládat ani fyzická zařízení, ta jsou řízena svým řídicím algoritmem. V manuálním režimu je naopak vše v roli operátora, ale to slouží pouze pro údržbu. V poloautomatickém režimu může operátor ručně provést akci ke splnění přechodu v dané fázi.

3.2 Simulátor

Pasterizační jednotka je realizována pomocí simulátoru na obrázku 3.2, který se skládá z tanku, napouštěcího a vypouštěcího ventilu, mixéru a ohřevu. Proces je ovládán ze softwaru TwinCAT, kde jsou implementovány programové bloky pro jednotlivé řídicí moduly a fáze dle normy ISA-88. Řízení, které by mělo být z nadřazeného systému, tedy aplikace typu Batch, je zde realizováno v programovém bloku Test pomocí stavového automatu.

Cílem projektu je napustit tank mlékem, ohřát ho na teplotu potřebnou k pasterizaci a takto pasterizované mléko za stálého míchání vypustit k dalšímu zpracování. Ohřev mléka je řízen regulátorem PID nastavením jeho složek. Řídicí moduly ventilů ovládají jejich otevření a zavření, přepínání manuálního a automatického režimu a hlásí chybu spuštění ventilu, která nastane v případě, že do požadovaného času nepřijde odezva ze simulátoru o jeho otevření nebo zavření. U motoru a ohřevu ovládají řídicí moduly zapnutí a vypnutí, přepínání manuálního a automatického režimu a opět hlásí chybu spuštění. Jsou zde implementovány čtyři fáze, a to napouštění tanku na zadanou hladinu, ohřev na zadanou teplotu, míchání po zadaný čas a vypuštění celého tanku. Řídicí moduly i fáze jsou v TwinCATu vizualizovány.

Komunikaci se simulátorem provádí PLC na základě protokolu Modbus TCP. V komunikaci jsou realizovány 4 funkce Modbus, a to čtení diskretních vstupů, čtení vstupních registrů, zápis více výstupních cívek a zápis více registrů. Test komunikace se simulátorem lze z počítače provést díky programu qModMaster, což je nástroj pro testování Modbus komunikace při ladění průmyslových aplikací. Nastavením IP adresy, portu a Unit ID se lze k simulátoru jednoduše připojit.



Obr. 3.2: Simulátor Pasterizační jednotky

3.2.1 TwinCAT

TwinCAT (The Windows Control and Automation Technology) je softwarová platforma společnosti Beckhoff pro průmyslovou automatizaci. Je to programovací prostředí založené na standardu IEC 61131-3. TwinCAT je rozdělen na inženýrskou část, určenou pro programování a konfiguraci, a provozní část, která vykonává funkce v reálném čase. Pro ladění projektů využívá vývojové prostředí Microsoft Visual Studio,

díky čemuž zvyšuje flexibilitu při vývoji. TwinCAT umožňuje podporuje různé průmyslové protokoly a komunikaci s širokou škálou zařízení, což pomáhá k vytváření řídicích systémů v různých průmyslových odvětvích. Díky možnosti běhu na různém hardwaru je vhodný jak pro malé aplikace, tak i pro rozsáhlé výrobní linky. TwinCAT se skládá ze dvou hlavních komponent:

- **XAE (eXtended Automation Engineering)** - Je to inženýrská část, která běží na vývojovém počítači. Slouží pro vytváření, ladění a nahrávání PLC kódů, dále pro konfiguraci připojeného hardwaru a simulaci aplikací. Je integrované do Visual Studio nebo běží jako samostatný nástroj. Toto prostředí poskytuje nástroje pro správu celého projektu.
- **XAR (eXtended Automation Runtime)** - Tato komponenta běží na cílovém zařízení, tedy průmyslovém počítači, PLC nebo embedded systému. Slouží k provádění nahraného řídicího programu v reálném čase. Zajišťuje stabilní běh systémových i uživatelských funkcí, přičemž jednotlivým úlohám lze přiřadit různé jádro procesoru. [25]

Order No	License
TC1200	TC3 PLC
TF4100	TC3 Controller Toolbox
TF5100	TC3 NC I
TF6250	TC3 Modbus-TCP
TF6340	TC3 Serial-Communication
TF6701	TC3 IoT Communication (MQTT)
TF6705	TC3 IoT Communication (I/O)
TF6710	TC3 IoT Functions
TF6720	TC3 IoT Data Agent
TF6730	TC3 IoT Communicator

Obr. 3.3: Seznam použitých licencí

Ke vzájemné komunikaci těchto dvou komponent se využívá protokol ADS (Automation Device Specification). Je to protokol od společnosti Beckhoff pro efektivní a bezpečný přenos dat, příkazů a diagnostiky. Reprezentuje transportní vrstvu v rámci TwinCATu, takže lze implementovat na různé fyzické sítě. V praxi se nejčastěji používá ADS komunikace přes protokol TCP/IP nebo UDP/IP. Každé zařízení v síti má přiděleno jedinečné AMSNetID, které identifikuje cílové zařízení. Díky ADS lze z jednoho počítače obsluhovat a ladit více zařízení současně. [26]

Licence v TwinCATu umožňují přístup k jednotlivým funkcím a modulům podle potřeb programátora. Runtime licence jsou určeny pro běh projektu na cílovém zařízení, bez jejichž aktivace by nebylo možné projekt spustit. TwinCAT nabízí 7denní trial licenci, kterou lze opakovaně využívat. To je ideální pro testování nebo školní účely. Pro produkční provoz je nutné zakoupit plné runtime licence k legálnímu použití platformy.

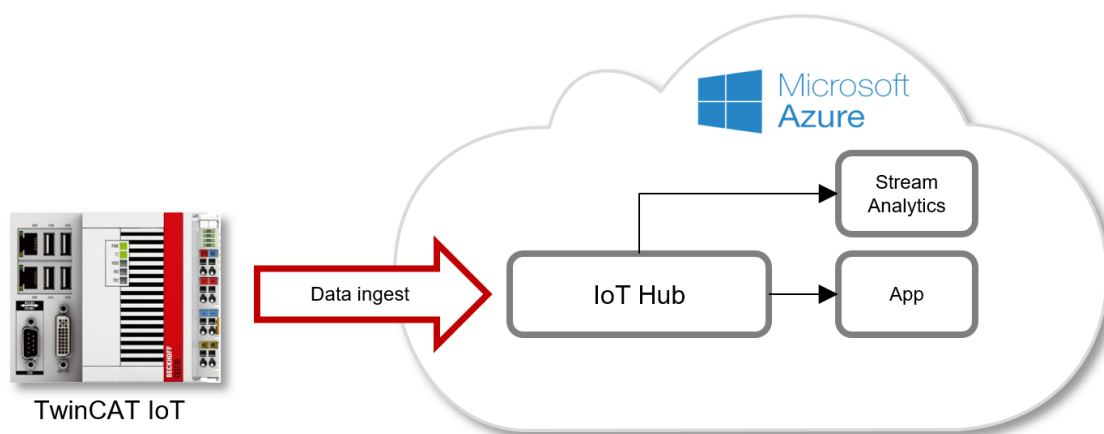
TwinCAT je kompatibilní s různými typy hardwaru a umožňuje spouštění jak na průmyslových počítačích, tak na embedded systémech nebo standardních počítačích. Integruje automatizační funkce, podporuje standardní komunikační protokoly a IoT technologie. V kombinaci s cloudovými službami, jako je Microsoft Azure, umožňuje monitorovat, analyzovat a řídit připojená fyzická zařízení.

4 Realizace digitálního dvojčete

Tato kapitola se věnuje realizaci digitálního dvojčete Pasterizační jednotky. Simulátor odesílá data do služby IoT Hub, odkud je odebírá služba Azure Digital Twins, která na jejich základě průběžně aktualizuje vlastnosti digitálního dvojčete. Současně je implementována i zpětná komunikace, která zajišťuje, že veškeré změny provedené v digitálním dvojčeti jsou odesílány zpět do TwinCATu, kde slouží k řízení simulátoru.

4.1 Integrace TwinCATu s Azure IoT Hubem

Z hlediska protokolu MQTT je TwinCAT v roli publishera, který odesílá data na server. IoT Hub zde vystupuje v roli brokera, který předává data subscriberům. V roli subscriberu může být libovolná služba, která zpracovává data na konkrétním tématu. Subscriberem může být zpětně také TwinCAT, čímž může předávat řídicí příkazy.



Obr. 4.1: Datový tok z TwinCATu do Microsoft Azure [27]

4.1.1 Vytvoření IoT Hubu

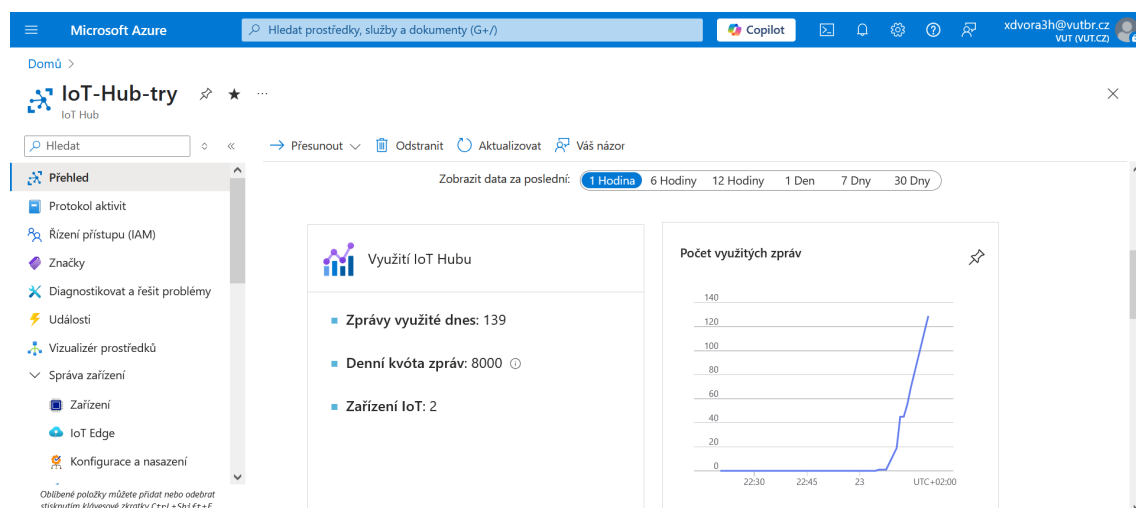
Prvním krokem pro realizaci bylo v platformě Microsoft Azure vytvoření Azure IoT Hubu, který zpracovává přijaté zprávy ze zařízení. Na domovské stránce Azure portálu, viz. obrázek 2.1, je možnost jednoduše vytvářet prostředky, neboli instance, zvolené služby. Lze to provést volbou Vytvořit prostředek nebo vyhledáním názvu dané služby v horní liště obrazovky. Předpokladem pro vytvoření služby je skupina prostředků. Skupina prostředků je kontejner, který obsahuje v jedné skupině všechny služby související s daným řešením.

Při vytváření jakékoli instance dané služby se postupně nastavuje několik parametrů. Parametry jsou rozděleny do záložek, přičemž většina služeb obsahuje tyto čtyři záložky:

- **Základy** - Obsahují základní nastavení služby, jako je volba předplatného, přiřazení do skupiny prostředků, volba oblasti a název instance služby. Dále se pak zadávají základní specifikace vytvářené služby.
- **Sítě** - Tato záložka slouží k nastavení přístupu k dané službě. Nejčastěji se definuje, zda-li bude prostředek dostupný veřejně nebo privátně.
- **Značky** - Zde se nastavuje dvojice název-hodnota. Slouží ke kategorizaci a přehlednosti všech prostředků, což se využije pouze při komplexnějších řešeních.
- **Zkontrolovat a vytvořit** - Tato záložka je posledním krokem, kde se již nic nenastavuje. Zde se automaticky kontrolují zadané parametry a v případě, že jsou v pořádku se služba vytvoří.

Další záložky, které se mohou zobrazit při vytváření služby, jsou rozdílné pro každou službu, a tak je nelze zobecnit. V této práci jsou při popisu vytváření jednotlivých služeb vždy popsány pouze důležité parametry, bez kterých by dané řešení nefungovalo.

V případě IoT Hubu je důležitým parametrem v záložce Základ denní limit přijatých zpráv a zvolený způsob připojení v záložce Síť. Je možnost zvolit buď veřejný přístup, nebo privátní přístup. Privátní přístup slouží pro produkční řešení, kdy se připojení k IoT Hubu provádí z virtuální interní sítě pomocí služby Azure Private Link. Pro účel této práce byl dostačující bezplatný limit 8000 zpráv za den a připojení přes veřejný přístup. Ostatní parametry lze ponechat defaultně nastavené.



Obr. 4.2: Přehled využití IoT Hubu

Vytvořený IoT Hub v sekci Přehled graficky znázorňuje jeho využití, tedy počet přijatých zpráv a připojených zařízení. V sekci Správa zařízení a podsekcí Zařízení se přidává nové zařízení. To slouží k identifikaci zařízení, které se snaží posílat data na cloud. Při jeho registraci se volí typ ověřování z hlediska bezpečnosti, jelikož IoT Hub vždy vyžaduje autentizaci a autorizaci. Nejčastější volbou je ověřování přes symetrický klíč. V sekci Nastavení zabezpečení a podsekcí Zásady sdíleného přístupu se nacházejí primární připojovací řetězce, které umožňují přístup k IoT Hubu. Rozdělují se podle zvoleného oprávnění – od plného přístupu, přes připojení služby či zařízení až pouze ke čtení registru.

IoT Hub slouží jako MQTT broker, který přijímá zprávy od publisherů. Samotný IoT Hub zprávy nezobrazuje, pouze je uchovává a předává dál subscriberům. Pro jejich zobrazení lze použít nástroj Azure IoT Explorer, který se připojuje k IoT Hubu pomocí připojovacího řetězce. Tento nástroj umožňuje sledovat příchozí telemetrii v reálném čase. Pokud se použije řetězec s oprávněním iothubowner, získá uživatel plný přístup a může v sekci Telemetry sledovat doručená data. Tím lze ověřit, že zařízení zprávu skutečně odeslalo a že byla úspěšně doručena do cloudu, podobně jako u subscriberů, kteří zprávy odebírají.

4.1.2 Autentizace klienta

K ověření identity klienta při navazování spojení s IoT Hubem lze využít SAS token nebo certifikáty X.509. SAS token je časově omezený digitální klíč, který je možné vygenerovat dvěma způsoby. První způsob je pomocí nástroje Azure IoT Explorer. V něm se výběrem příslušného zařízení v sekci Device identity objeví možnost vygenerování připojovacího řetězce obsahujícího potřebný SAS token. Před samotným generováním je nutné zvolit, zda bude token založen na primárním nebo sekundárním symetrickém klíči. Tato volba závisí na uživateli, přičemž z hlediska funkce není mezi klíči rozdíl. Nastavit se musí i doba platnosti SAS tokenu pro zajištění stabilního spojení. Vygenerovaný připojovací řetězec následně obsahuje Hostname, DeviceId a požadovaný SAS token.

Druhý způsob vyžaduje instalaci Azure CLI, které vlastně jen rozšiřuje běžný příkazový řádek o instrukce ke správě IoT Hubu. Tento nástroj je dostupný pro Windows, macOS a Linux, kdy usnadňuje spouštění hromadných úkonů a je kompatibilní s většinou Microsoft Azure služeb. Jak ukazuje obrázek 4.3, tak nejprve je zapotřebí se přihlásit ke službě Microsoft Azure. Po přihlášení pak stačí zadat jednoduchý příkaz obsahující základní informace o IoT Hubu a dobu platnosti, načež se vygeneruje SAS token. Výhodou tohoto způsobu je možnost automatizace generování tokenů skriptem, například při nasazení většího počtu zařízení. Zároveň je možné sledovat příchozí zprávy bez nutnosti otevírat grafické rozhraní.

```

C:\Users\tomas>az login
Select the account you want to log in with. For more information on login with Azure CLI, see https://go.microsoft.com/fwlink/?linkid=2271136

Retrieving tenants and subscriptions for the selection...

[Tenant and subscription selection]

No      Subscription name      Subscription ID      Tenant
-----
[1] *    Azure subscription 1  08fd1023-748a-4106-a3e6-8ac0879305ea  VUT

The default is marked with an *; the default tenant is 'VUT' and subscription is 'Azure subscription 1' (08fd1023-748a-4106-a3e6-8ac0879305ea).

Select a subscription and tenant (Type a number or Enter for no changes):

Tenant: VUT
Subscription: Azure subscription 1 (08fd1023-748a-4106-a3e6-8ac0879305ea)

[Announcements]
With the new Azure CLI login experience, you can select the subscription you want to use more easily.
Learn more about it and its configuration at https://go.microsoft.com/fwlink/?linkid=2271236

If you encounter any problem, please open an issue at https://aka.ms/azclibug

[Warning] The login output has been updated. Please be aware that it no longer displays the full list
of available subscriptions by default.

C:\Users\tomas>az iot hub generate-sas-token --hub-name IoT-Hub-try.azure-devices.net --device-id DalsiZarizeni --duration 3600
{
  "sas": "SharedAccessSignature sr=IoT-Hub-try.azure-devices.net%2Fdevices%2FDalsiZarizeni&sig=UYhtCH2YAAWFpRQpnQj0MfN8kEbRYuW%2B7aDIHE%2BDSv4%3D&se=1747776104"
}

```

Obr. 4.3: Generování SAS tokenu pomocí Azure CLI

Pro využití certifikátů X.509 je nutné vygenerovat privátní klíč a certifikát klienta. K tomu je zapotřebí certifikačních nástrojů, kdy nejčastější volbou je software OpenSSL, který slouží jako nástroj pro kryptografické operace a generování šifrovaných certifikátů. S využitím aplikace Git Bash je možné vygenerovat a uložit do počítače privátní klíč a certifikát klienta.

Pro oba způsoby ověření klienta je nezbytný také kořenový certifikát DigiCert Global Root G2. Tento certifikát, vydaný certifikační autoritou DigiCert, se používá k bezpečnosti a důvěryhodnosti komunikace mezi klientem a serverem, v tomto případě mezi TwinCATem a IoT Hubem. Stáhnout ho lze přímo ze svého počítače a to v případě operačního systému Windows z Microsoft Management Console. Zadáním příkazu mmc.exe se otevře konzole, v níž je třeba v sekci Soubor vybrat volbu Přidat nebo odebrat modul snap-in. Následně vybrat Certifikáty a přidat je přes uživatelský účet, aby bylo možné procházet obsah úložišť. V levém okně se objeví všechny dostupné certifikáty, přičemž v sekci Důvěryhodných kořenových certifikačních autorit se musí vyhledat certifikát DigiCert Global Root G2 a exportovat ho do svého počítače. Certifikáty mohou být uloženy v různých formátech, ale nejčastěji se využívá PEM (Privacy Enhanced Mail) nebo DER (Distinguished Encoding

Rules). Formát PEM se používá zejména pro webové servery, kdy se jedná o textový soubor zakódovaný v Base64 ASCII se specifickou hlavičkou. Naopak DER je binární soubor a používá se v prostředí Java. Správně nakonfigurované certifikáty zvyšují zabezpečení přenosu dat. Certifikáty mají omezenou platnost, tudíž je důležitá jejich včasná obnova, aby se předešlo výpadkům komunikace. [28]

4.1.3 Publikování dat

TwinCAT nabízí balíček TF6701 zaměřený na IoT funkce, který umožňuje naimportovat knihovnu Tc3_IotBase. Tato knihovna obsahuje mimo jiné funkční blok IotMqttClient, který slouží k implementaci MQTT klienta pro komunikaci s MQTT brokerem. V této knihovně se nachází funkční blok IotMqttClient, který implementuje MQTT klienta pro komunikaci s brokerem. Před spojením s IoT Hubem byl proveden test komunikace TwinCATu se serverem skupra.cz, fungujícím jako Mosquitto broker. Odesláním jednoduché zprávy bylo ověřeno, že komunikaci přes protokol MQTT nic neblokuje a TwinCAT je schopen navázat spojení s IoT Hubem, který klade vyšší požadavky na zabezpečení a autentizaci zařízení. Funkční blok obsahuje proměnnou bConnected, která indikuje úspěšné navázání spojení.

```
PROGRAM MAIN
VAR
    sTopicPub    : STRING(255) := 'devices/Simulator_tanku/messages/events/readpipe'; //tema publish
    sTopicSub    : STRING(255) := 'devices/Simulator_tanku/messages/devicebound/#'; //tema subscribe
END_VAR

IF bSetParameter THEN
    bSetParameter := FALSE;
    fbMqttClient.stTLS.sCA := 'C:\Certs\digicertG2.cer'; //lokal
    fbMqttClient.stTLS.bNoServerCertCheck := FALSE;
    fbMqttClient.stTLS.sVersion := 'tls1.2';
    fbMqttClient.nHostPort := 8883;
    fbMqttClient.sTopicPrefix := '';
    fbMqttClient.sHostName := 'IoT-Hub-try.azure-devices.net';
    fbMqttClient.sClientId := 'Simulator_tanku';
    fbMqttClient.sUserName := 'IoT-Hub-try.azure-devices.net/Simulator_tanku/?api-version=2021-04-12';
    fbMqttClient.sUserPassword := 'SharedAccessSignature sr=IoT-Hub-try.azure-devices.net%2Fdevices%2FSimulator_tanku&sig=7YafpXxGW5ZrdcuM20mOV0Cx6f37xvSGrRmS1m7GmTc%3D&se=1747651540';
    ;
    fbMqttClient.ipMessageQueue := fbMessageQueue; //subscribe
END_IF
```

Obr. 4.4: Parametry pro připojení k IoT Hubu

Po ověření schopnosti odeslání dat z TwinCATu a získání SAS tokenu nebo certifikátů X.509, bylo nutné pro spojení s IoT Hubem upravit parametry funkčního bloku. Microsoft Azure vyžaduje pro oba způsoby lehce odlišné parametry. Na obrázku 4.4 jsou zobrazeny parametry pro připojení k IoT Hubu pomocí SAS tokenu. Ačkoliv vygenerovaný připojovací řetězec se SAS tokenem z Azure IoT Exploreru již obsahuje informace o IoT Hubu i konkrétním zařízení, tak je nutné tyto údaje

explicitně definovat v parametrech. Důležité je definovat cestu ke kořenovému certifikátu a aktivovat kontrolu platnosti tohoto certifikátu nastavením proměnné `bNoServerCertCheck` na `FALSE`. Nastavení hodnoty na `TRUE` znamená, že není tento certifikát kontrolován, což slouží pouze pro účely testování a neumožňuje zaručit bezpečnost, jelikož by se umožnilo připojení i bez platného certifikátu.

Microsoft Azure vyžaduje použití protokolu TLS verze 1.2 a připojení přes šifrovaný port 8883, který tento protokol využívá. Parametry jako `Hostname`, `ClientId` a `UserName` s verzí API (Application Programming Interface) slouží pro identifikaci serveru a klienta. Tyto údaje lze najít v Azure portálu v sekci Přehled v IoT Hubu. Posledním parametrem je pak vygenerovaný SAS token. V deklaraci proměnných je ještě zapotřebí správně nastavit téma. Tímto nastavením bylo zajištěno bezpečné a autorizované odesílání dat z TwinCATu do Azure IoT Hubu přes protokol MQTT. [29, 30, 31]

Tue May 20 2025 23:08:28 GMT+0200 (Středoevropský letní čas):

```
{
  "body": "{\"temperature\": 22.03053, \"level\": 1.140191,
  \"filling_valve_is_closed\": TRUE, \"filling_valve_is_opened\": FALSE,
  \"draining_valve_is_closed\": TRUE, \"draining_valve_is_opened\": FALSE,
  \"heating_state\": FALSE, \"mixing_state\": TRUE}\u0000",
  "enqueuedTime": "Tue May 20 2025 23:08:28 GMT+0200 (Středoevropský letní čas)"
  "properties": {
    "readpipe": ""
  }
}
```

Tue May 20 2025 23:08:17 GMT+0200 (Středoevropský letní čas):

```
{
  "body": "{\"temperature\": 21.74841, \"level\": 1.153356,
  \"filling_valve_is_closed\": TRUE, \"filling_valve_is_opened\": FALSE,
  \"draining_valve_is_closed\": TRUE, \"draining_valve_is_opened\": FALSE,
  \"heating_state\": FALSE, \"mixing_state\": TRUE}\u0000",
  "enqueuedTime": "Tue May 20 2025 23:08:17 GMT+0200 (Středoevropský letní čas)"
  "properties": {
    "readpipe": ""
  }
}
```

Obr. 4.5: Příchozí zprávy do IoT Hubu

Na obrázku 4.5 jsou vidět zprávy odeslané z TwinCATu do IoT Hubu. Každá zpráva se skládá z těla, časové značky a vlastností. Tělo zprávy se tvoří v TwinCATu, kdy se odesílá jako JSON objekt. Hodnoty ze simulátoru jsou převedeny do stringu a následně spojeny za sebou.

4.2 Digitální model v Microsoft Azure

Digitální model je základem digitálního dvojčete. Služba Azure Digital Twins umožňuje definovat strukturu, vlastnosti a propojení fyzického systému. Dobře navržené digitální dvojče odráží aktuální stav, zajišťuje větší přehlednost a podporuje efektivní řízení a údržbu. V této kapitole je popsáno vytvoření 3D modelu v Siemens NX a jeho následné využití v Azure Digital Twins.

4.2.1 Vytvoření Azure Digital Twins

Pro vytvoření služby Azure Digital Twins je nutné vybrat příslušnou skupinu prostředků, zadat název, vybrat vhodnou oblast a přiřadit roli Vlastníka dat služby Azure Digital Twins. Bez této role by uživatel neměl oprávnění k práci s rozhraním API pro správu digitálních dvojčat. Oprávnění lze kdykoliv měnit v sekci Řízení přístupu (IAM), kdy volbou Přidat lze přiřadit role jakémukoliv uživateli nebo jakékoliv službě. Dále se volí metoda připojení, kdy se obdobně jako u IoT Hubu volí buď veřejný koncový bod, nebo privátní koncový bod.

V sekci Přehled lze sledovat využití zpráv, každopádně pro lepší přehled slouží sekce Sledování. Zde je možné zobrazit metriky, kde lze sledovat veškeré statistiky o digitálních dvojčatech, například počet zaslaných zpráv nebo požadavků. K vytvoření digitálních dvojčat se využívá nástroj Azure Digital Twins Explorer, který je dostupný přímo z Azure portálu. Přechází se do něj ze sekce Přehledu, kdy je v horní liště volba Otevřít Azure Digital Twins Explorer (náhled).

Po jeho spuštění je potřeba zadat URL adresu služby tak, že se před název hostitele služby Azure Digital Twins napíše předpona `https://`. Název hostitele služby je zobrazen v sekci Přehled. Digitální dvojče se vytváří z modelu, který se vkládá v jazyce DTDL v záložce Models. Vložený model, viz. obrázek 4.6, by měl popisovat vlastnosti a obsahovat definici vztahů s ostatními modely. Tyto vztahy jsou pak automaticky vizualizovány v záložce Model graph, která slouží k pochopení architektury, což se využije zejména u komplexních systémů.

Digitální dvojče se vytvoří kliknutím na příslušný model a volbou Create a Twin. Každé dvojče je třeba pojmenovat tak, aby bylo jasně identifikovatelné v celé digitální replice. Z jednoho modelu lze vytvořit nekonečno mnoho instancí digitálních dvojčat, které lze vidět v záložce Twins.

Po kliknutí na konkrétní dvojče se v pravé části obrazovky zobrazí hodnoty jeho vlastností, včetně metadat, což je čas poslední aktualizace jednotlivých vlastností. V záložce Twin Graph jsou zobrazena dvojčata v přehledné grafické podobě. Právě zde je možné mezi nimi vytvářet vztahy na základě definic v modelech, čímž se propojuje jejich vzájemná logika.

```

{
  "@id": "dtmi:example:Valve;1",
  "@type": "Interface",
  "displayName": "Valve",
  "@context": "dtmi:dtdl:context;2",
  "contents": [
    {
      "@type": "Property",
      "name": "valve_opened",
      "schema": "boolean"
    },
    {
      "@type": "Property",
      "name": "valve_closed",
      "schema": "boolean"
    }
  ]
}

```

Obr. 4.6: Model Valve v jazyce DTDL

Každý vztah musí mít unikátní název, jinak vznikne chyba. Tento vztah se vytvoří tak, že se nejprve vybere výchozí dvojče a následně se vybere kliknutím pravého tlačítka myši to dvojče, se kterým má být propojeno. Díky tomu se zobrazí volba Add relationships, která umožňuje daný vztah vytvořit. Tímto způsobem vzniká graf digitálních dvojčat, jak je znázorněno na obrázku 2.2. Hodnoty vlastností dvojčat může upravovat kdokoli, kdo má přiřazený přístup. Nezmění-li nějakou hodnotu vlastnosti digitálního dvojčete manuálně uživatel, je potřeba pro její zobrazení spustit dotaz tlačítkem Run Query, které spouští SQL dotazy nad dvojčaty.

4.2.2 Vytvoření modelu digitálního dvojčete

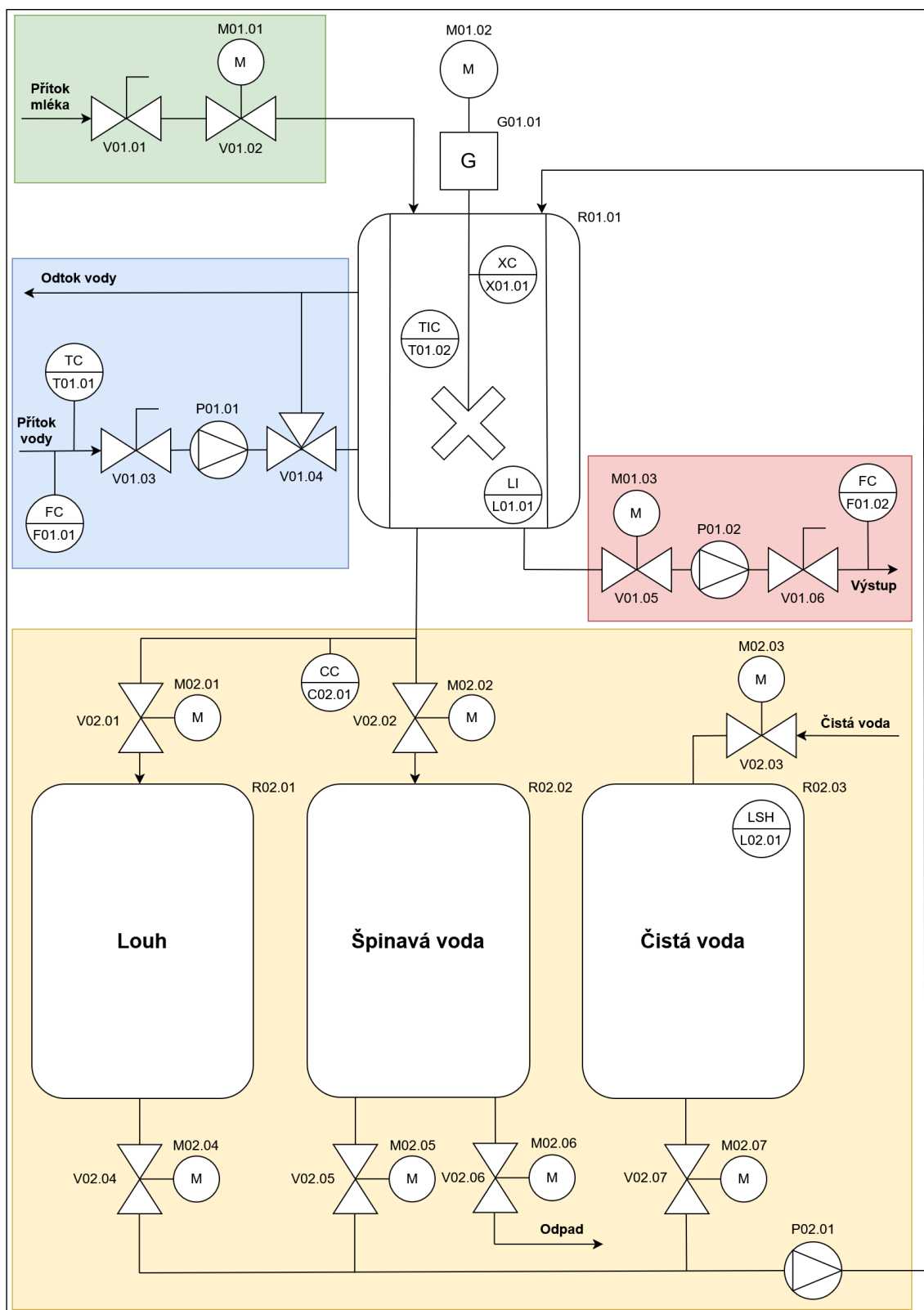
Základem pro vytvoření digitálního modelu byl P&ID (Piping and Instrumentation Diagram) diagram znázorňující celou Pasterizační jednotku. Tento diagram byl navržen na základě simulátoru a následně byl rozšířen o fázi čištění. Pro přiblížení skutečnému provozu byl dále doplněn o potřebné komponenty, jako jsou ventily, snímače a potrubí, které se vyskytují v reálném technologickém procesu. Diagram na obrázku 4.8 zachycuje následující části technologie:

- **Hlavní tank** - Tank R.01.01 je vybaven míchadlem s motorem M01.02 s převodovkou G01.01, snímačem teploty T01.02, snímačem hladiny L01.01 a snímačem otáček X01.01.
- **Přívod mléka (zelená oblast)** - Obsahuje ruční ventil V01.01 a elektrický ventil V01.02.
- **Přívod a odtok vody (modrá oblast)** - obsahuje čerpadlo P01.01, ventily V01.03 a V01.04, průtokoměr F01.01 a teplotní čidlo T01.01.
- **Výstup pasterizace (červená oblast)** - Tvořen elektrickým ventilem V01.05, čerpadlem P01.02, ručním ventilem V01.06 a průtokoměrem F01.02.
- **Čištění (žlutá oblast)** - Zahrnuje tank na louh R02.01, špinavou vodu R02.02 a čistou vodu R02.03, které jsou vybaveny elektrickými ventily na přívodu i odvodu. Dále obsahuje snímač koncentrace C02.01, snímač hladiny L02.01 a čerpadlo P02.01.

Pro tvorbu 3D modelu Pasterizační jednotky byl zvolen software Siemens NX, který slouží pro detailní a virtuální reprezentaci výrobních celků. Je to v praxi široce využívaný program podporující paralelní spolupráci více odvětví, čímž lze vytvářet náročné mezioborové projekty. Vyjma komplexního a intuitivního 3D modelování obsahuje také moduly pro simulaci (MCD), kinematické analýzy nebo návrh elektrických systémů. [32]

Na trhu existuje řada podobných nástrojů pro 3D modelování, jako je například SolidWorks, kde každý z nich je specifický pro dané řešení. V rámci této práce však byl zvolen právě Siemens NX, a to především z důvodu předchozí praktické zkušenosti s tímto nástrojem. Díky tomu nebylo nutné věnovat čas studiu jeho rozhraní a funkcí, a tak bylo možné se soustředit rovnou na samotný návrh modelu. Cílem nebylo pouze vytvoření 3D modelu Pasterizační jednotky, ale zejména jeho následné využití ve službě Azure Digital Twins, konkrétně v nástroji 3D Scenes pro vizualizaci digitálního dvojčete.

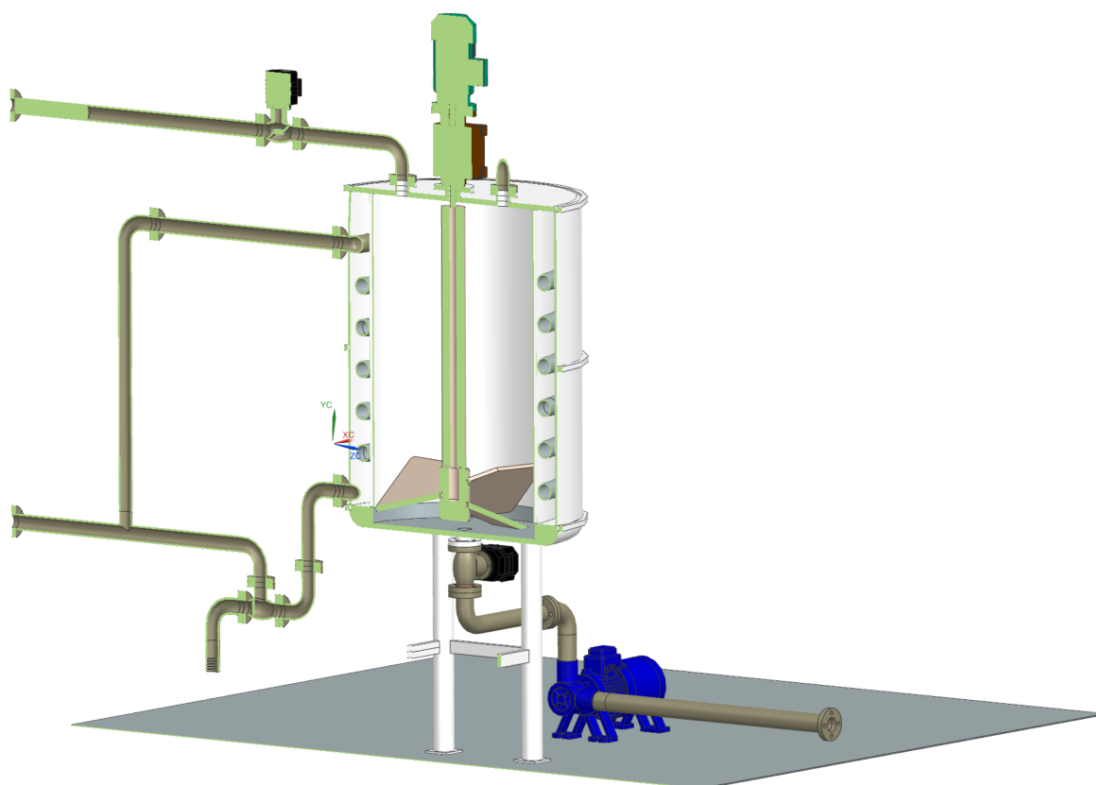
Modelování bylo rozděleno do dvou hlavních sestav. První z nich, pojmenovaná `simulator_tanku`, představovala digitální kopii simulátoru Pasterizační jednotky. Zahrnovala modely ventilů, motorů, převodovky, tanku, mixéru a ohřevu. Smyslem bylo vytvořit model co nejvíce podobný odpovídající reálné výrobní lince, a proto k němu bylo přidáno veškeré potrubí s čerpadly, která simulují průběh technologického procesu. Druhá sestava s názvem `cisteni` reprezentovala čištění skládající se ze tří tanků - s louhem, špinavou vodou a čistou vodou. Opět k nim byla doplněna veškerá potrubí, ventily a čerpadlo. Pro lepší přehlednost byly jednotlivé díly barevně odlišeny.



Obr. 4.7: P&ID diagram Pasterizační jednotky

Obě vytvořené sestavy byly následně vloženy do finální sestavy s názvem `final_model_pasterizacni_jednotka`. Tato sestava je koncipována tak, že první sestava byla vložena dvakrát, což simuluje reálné výrobní prostředí s několika identickými stanicemi. Tento přístup tak lze označit jako modulární, který je výhodný zejména při případném rozšiřování výroby, respektive sestav. Oproti tomu byla druhá sestava vložena pouze jednou, jelikož se předpokládá, že čištění bude sdílené pro všechny tanky, což je z hlediska ekonomiky výhodné. Mezi tyto sestavy bylo vloženo propojovací potrubí, čímž vznikl uzavřený okruh, ve kterém může probíhat pasterizace včetně automatizovaného čištění.

Modelování v Siemens NX není principiálně složité, ale mnohdy je velmi časově náročné. Velký důraz musí být kladen na geometrickou přesnost a dodržení správného napojení jednotlivých dílů k sobě. Každý díl musí být vytvořen ručně, ale v některých případech lze využít internetovou knihovnu. Do sestavy se musí každý díl rovněž ručně vložit a připojit vhodnými vazbami. Při spojení obou sestav byla zásadní správná výška komponentů, kdy jen malá odchylka způsobila nepřesnost a nebylo možné je potrubím propojit.



Obr. 4.8: Řez hlavního tanku v Siemens NX

Při modelování tanku byla výzva namodelovat dvojitý plášť, v němž se nachází trubka pro ohřev mléka. Bylo nutné pracovat s posunutím referenčních rovin, jelikož potrubí obepíná tank pod určitým úhlem. Musela se tak správně vytvořit díra v tanku pro napojení přívodu a odvodu vody. Celý model vznikl postupně, kde byly jednotlivé komponenty postupně navrhovány a upravovány. U každého dílu je vyjma barvy možné definovat jeho materiál. U jednotlivých dílů může být z knihovny přiřazen vyhovující materiál, nicméně není okem vidět a nemá přímý vliv na celkové řešení. Model by se mohl rozšířit o snímače hladiny, teploty nebo průtoku, které jsou v P&ID diagramu navrženy, čímž by se posílila věrnost zobrazení reálného provozu.

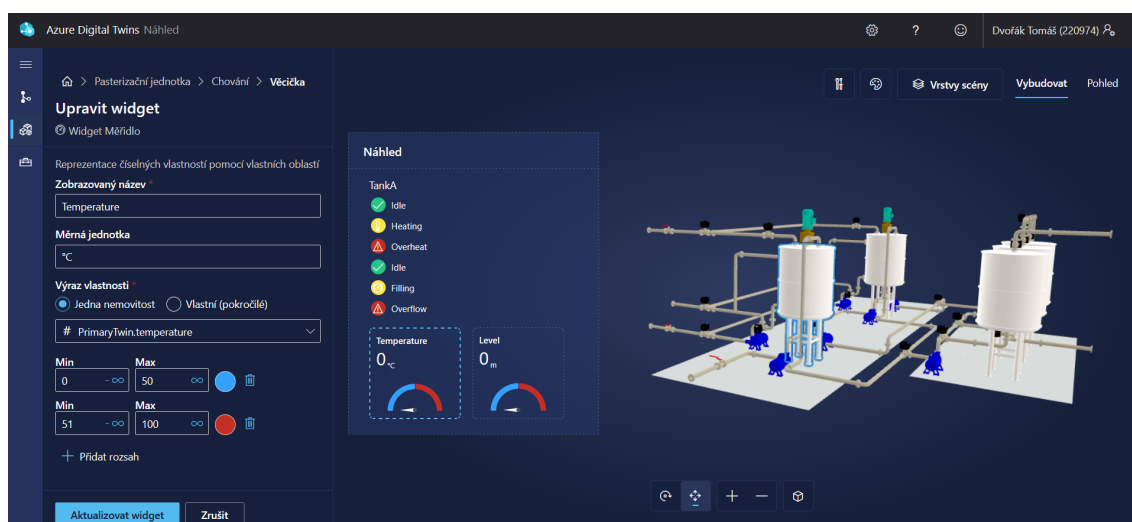
4.2.3 Propojení s 3D Scenes

Pro využití 3D modelu Pasterizační jednotky v prostředí 3D scenes je nutné, aby byl model ve formátu .glb nebo .gltf, což jsou formáty vhodné pro efektivní přenos 3D modelů. Export modelu v tomto formátu není v Siemens NX podporován, tudíž bylo potřeba využít alternativní způsob konverze.

Model byl nejprve exportován ve formátu .obj, který zachovává celou geometrii objektu. Pomocí online konvertorů bylo možné převést formát .obj na formát .glb, kdy výsledný model odpovídal původnímu modelu v Siemens NX. Problém však nastal ve chvíli, kdy byl model nahrán do prostředí 3D Scenes. Při následném označení jednoho dílu, například jednoho tanku, se automaticky označily i všechny ostatní. Bylo to nejspíše zapříčiněno tím, že všechny tanky byly vytvořeny ze stejného výchozího dílu a ačkoliv byly uloženy s jiným názvem, tak při konverzi došlo k jejich spojení do jednoho objektu. Bylo tak potřeba rozdělit geometrii modelu, aby byl každý díl reprezentován samostatně. K tomu byl zvolen software Blender, jenž obsahuje nástroje pro úpravu 3D modelů a nabízí export do požadovaného formátu. Pomocí jeho funkcí byly jednotlivé díly odděleny a připraveny pro export.

Tento problém ukazuje, že pouhý přenos modelu mezi dvěma softwary může být občas poměrně komplikovaný, i když se jedná o běžný způsob reprezentace 3D modelu. Potvrzuje se tak, že standardizace není v digitálním světě klíčová pouze při komunikaci mezi systémy.

Před samotným nahráním 3D modelu je potřeba nejprve 3D Scenes inicializovat. To zahrnuje zadání URL hostitele, podobně jako u grafu digitálních dvojčat, a URL adresy úložiště Azure storage, jež bylo vytvořeno při publikování Azure funkce. Po této konfiguraci se tlačítkem Přidat nové vytváří scéna pro 3D model. Zadáním názvu scény a jejím volitelným popisem je možné model ve formátu .glb nahrát a následně zobrazit. 3D Scenes nabízí dva režimy, kdy první z nich slouží pro úpravy a druhý pro zobrazení. Pro detailní zobrazení jednotlivých částí modelů lze pomocí myši celou scénu volně otáčet, různě posouvat, přibližovat nebo oddalovat.

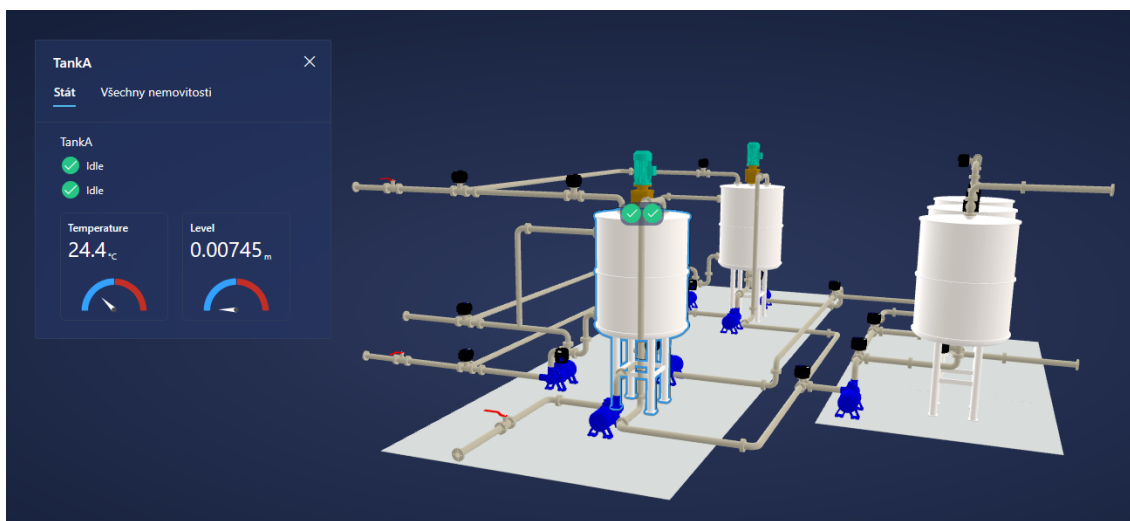


Obr. 4.9: Editace widgetů v 3D Scenes

V editaci nabízí 3D Scenes na levé straně intuitivní grafické rozhraní. V záložce Elements se volbou Nový prvek označí jeden nebo více dílů, kterým lze přiřadit ID příslušného digitálního dvojčete vytvořeného v Azure Digital Twins. Dochází tak k propojení datového toku mezi fyzickým zařízením a digitálním modelem. Tomuto prvku pak lze v záložce Chování přiřadit chování ve formě vizuálních pravidel nebo widgetů. Pravidla umožňují definovat různé vizuální stavy na základě dat digitálního dvojčete. Například u tanku lze nastavit několik teplotních rozmezí, pro která se automaticky změní barva tanku nebo se objeví zvolená ikona. Díky tomu lze snadno získat vizuální přehled o aktuálním stavu teploty a rychle rozpoznat případné odchylky nebo problémy v provozu. Widgety oproti tomu slouží pro zobrazení aktuálních hodnot vlastností za pomoci indikátorů.

V pravé horní části lze 3D modelu přidat různý styl, barvu nebo měnit pozadí scény. Vedle toho se nachází volba pro přepínání režimu zobrazení, kdy v případě přepnutí do režimu Pohled se nastavené chování v reálném čase promítá do 3D prostředí. Při komplexnější úloze vzniká interaktivní 3D prostředí poskytující přehled o celé výrobě.

Na obrázku 4.10 je znázorněno digitální dvojče Pasterizační jednotky. Jak lze vidět, tak integrace digitálního modelu ze softwaru Siemens NX do prostředí 3D Scenes přináší přehlednou vizualizaci, která umožňuje rychlou identifikaci provozních stavů i potenciálních problémů v reálném čase.



Obr. 4.10: Digitální dvojče v 3D Scenes

4.3 Zpracování a přenos dat

Efektivní zpracování a přenos dat jsou klíčovými prvky v každé průmyslové výrobě. V této kapitole je detailně popsán způsob, jak probíhá komunikace mezi fyzickým zařízením a digitálním dvojčetem, včetně ukládání dat do databáze. Byla využita Azure funkce, která umožňuje automatizovaný příjem dat a zpětnou komunikaci.

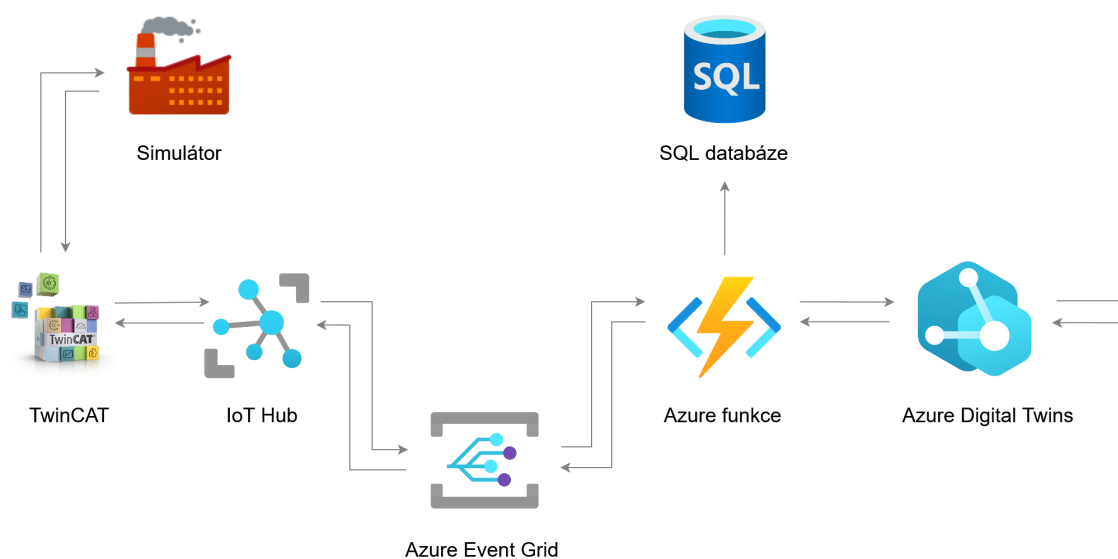
4.3.1 Zpracování dat pomocí Azure Funkce

Pro aktualizaci digitálního dvojčete ze služby IoT Hub byla využita Azure funkce. Azure funkce je řízená událostmi a poskytuje výpočetní prostředky pro Azure služby. Azure funkce přijme data z IoT Hubu a na základě přístupu ke službě Azure Digital Twins aktualizuje vlastnosti příslušného digitálního dvojčete.

Nejprve se vytvořil nový projekt ve Visual Studiu, kde se použila šablona pro Azure funkci v jazyce C#. Pro tento způsob bylo důležité nastavit trigger typu Event Grid. Na pozadí Microsoft Azure totiž funguje služba Azure Event Grid. Tato služba má za cíl distribuovat události mezi jednotlivými službami. Je přímo navržena pro reakci na změny, jako je nový soubor v úložišti, změna vlastnosti digitálního dvojčete nebo příchozí zpráva z IoT Hubu. Funguje na principu publisher-subscriber, kde jedna služba vygeneruje událost a jiná služba v roli subscribera na ni reaguje. Důležitou roli v tom hraje trigger, který spouští službu, jakmile vznikne událost. Azure Event Grid tak nabízí asynchronní spouštění událostí, aniž by se daná služba musela neustále dotazovat. [33]

Pro programování Azure funkcí se využívají knihovny, jenž se přidávají ve formě NuGet balíčků. Jsou to knihovny třetích stran, v tomto případě Microsoftu, umožňující různé ověřování nebo přístup ke službám. Instalují se do projektu pomocí NuGet Package Manager v záložce Project. Do prostředí se pak přidávají direktivou using, což umožní používat jejich třídy a metody.

Azure funkce využívá asynchronní metodu s parametrem CloudEvent, který obsahuje JSON objekt se zprávou z IoT Hubu. Tělo této zprávy je postupně zpracováno tak, aby se z něj převzaly správné hodnoty. Tyto hodnoty jsou poté předávány příslušným digitálním dvojčatům služby Azure Digital Twins. Takto vytvořená Azure funkce musí být jako celý projekt publikována do Microsoft Azure.



Obr. 4.11: Blokové schéma procesu

V rámci publikování se nejprve vybírá cíl, kterým je platforma Azure, a následně se specifikuje konkrétní typ cíle, což je v tomto případě aplikace funkcí. Tuto aplikaci lze vytvořit v Azure portálu nebo automaticky během procesu publikace z Visual studia. Volbou Create new se zadává název aplikace, vybírá se příslušná skupina prostředků, typ hostovacího plánu a oblast. Součástí je i zřízení trvalého úložiště Azure Storage pro provoz funkce a Application Insights pro monitorování funkce.

Po úspěšném publikování se v Azure portálu vytvoří aplikace funkcí. V sekci Přehled se zobrazí všechny Azure funkce publikovaného projektu. Citlivé informace, jako jsou připojovací řetězce, přihlašovací údaje nebo URL adresy hostitele, se v Azure funkci definují jako statické proměnné. V sekci Nastavení a podsekci Proměnné prostředí se tyto hodnoty ukládají z důvodu bezpečnosti, flexibility řešení a snadnější údržby.

Pro příjem událostí je zapotřebí vytvořit jejich odběr. Ten se nastavuje v IoT Hubu v sekci Události (Preview). Pro jeho vytvoření je nutné zadat název odběru, určit název systémového tématu služby Azure Event Grid, vybrat typ událostí, které budou spouštět odběr, a definovat koncový bod. Systémové téma Azure Event Gridu vzniká automaticky a díky němu je schopna tato služba přijímat události. Typem událostí jsou akce související se zařízeními a jejich telemetrií, tedy připojení, odpojení, změny stavu nebo přenos dat. Koncovým bodem, tedy místem pro zpracování událostí, je publikovaná Azure funkce spouštějící se vždy při přijetí události.

Dále bylo nutné zaregistrovat poskytovatele služby Azure Event Grid v předplatném Azure účtu. V sekci Nastavení a podsekci Poskytovatelé prostředků jsou zobrazeni dostupní poskytovatelé. Poskytovatel představuje rozhraní, které umožňuje provoz konkrétní služby. Pro každou službu musí být zaregistrován odpovídající poskytovatel, přičemž někteří jsou již automaticky zaregistrováni. Pro službu Azure Event Grid bylo nutné poskytovatele Microsoft.EventGrid ručně zaregistrovat.

Aby měla Azure funkce oprávnění pro přístup k libovolné Azure službě, je potřeba ji nastavit správnou roli. V aplikaci funkcí v sekci Nastavení a podsekci Identita je nutné zapnout systémově spravované identity a následně přiřadit roli. V tomto nastavení se pak pro danou skupinu prostředků přiřadí role Vlastník dat. Toto nastavení způsobí, že při příchozí události bude mít Azure funkce přístup k aktualizaci hodnot digitálních dvojčat služby Azure Digital Twins.

V Azure funkci je implementován konstruktor s parametrem logger, který slouží pro zaznamenávání běhu funkce. Tyto záznamy lze vidět v Azure funkci, respektive v jejím detailu, v záložce Volání. Jak ukazuje obrázek 4.13, tak tato záložka zobrazuje posledních 30 spuštění funkce. Jednotlivá spuštění obsahují podrobný výpis se všemi záznamy, včetně chybových hlášek, sloužících k ladění funkce.

Datum	Stav	Kód výsledku
20. 5. 2025 23:18:07	Úspěch	0
20. 5. 2025 23:17:56	Úspěch	0
20. 5. 2025 23:17:46	Úspěch	0
20. 5. 2025 23:17:35	Úspěch	0
20. 5. 2025 23:17:25	Úspěch	0
20. 5. 2025 23:17:14	Úspěch	0
20. 5. 2025 23:17:04	Úspěch	0
20. 5. 2025 23:16:53	Úspěch	0
20. 5. 2025 23:16:43	Úspěch	0
20. 5. 2025 23:16:32	Úspěch	0

Časové razítko	Typ	Zpráva
20. 5. 2025 23:18:07	Information	Executing 'Functions.Function1' (Reason='EventGrid trigger fired at 2025-05-20T21:18:07.1202850+00:00', Id=b0dd0706-8ead-41fb-ab3e-0bbf1b198a00)
20. 5. 2025 23:18:07	Information	ADT service client connection created.
20. 5. 2025 23:18:07	Information	Conversion to JSON data is successfully done.
20. 5. 2025 23:18:07	Information	Deserealization to JSON data is successfully done.
20. 5. 2025 23:18:07	Information	Temperature and level are updated.
20. 5. 2025 23:18:07	Information	States of filling valve are updated.
20. 5. 2025 23:18:07	Information	States of draining valve are updated.
20. 5. 2025 23:18:07	Information	Mixer state is updated.
20. 5. 2025 23:18:07	Information	Heater state is updated.

Obr. 4.12: Volání Azure funkce

4.3.2 Databáze

V každém výrobním podniku vzniká při provozu velké množství dat, které je nutno někam uložit a zpracovat. Nejčastěji se využívají relační databáze a nástroj Microsoft SQL Server. Řešení v platformě Microsoft Azure má tu výhodu, že databázový server běží na cloudu, takže není potřeba žádného fyzického hardwaru. Relační databáze ukládají data do přehledných tabulek, mezi nimiž lze definovat relace. Pomocí jazyka SQL je možné v nich snadno vyhledávat, filtrovat či dále zpracovávat. Přistupovat do ní může každý uživatel přes SQL server, pokud zná přihlašovací údaje. Toto řešení bylo použito v diplomové práci pro ukládání dat ze simulátoru.

Databáze SQL se jako ostatní služby musela vytvořit v Azure portálu. Vyžaduje zadat název, vybrat předplatné, příslušnou skupinu prostředků a především server. Ten je možné vytvořit v rámci tohoto procesu, kdy stačí zadat název, oblast a nastavit způsob ověřování pro přístup do databáze. Pro tento případ byl vybrán způsob ověřování přes SQL i přes Microsoft Entra, což vyžadovalo výběr správce serveru a nastavení přihlašovacích údajů. Další nastavení databáze pak souvisí s náklady podle využití. V rámci Azure je zdarma úložiště 32 GB dat/měsíčně, což je pro tuto práci dostačující. Nabídka na zdarma verzi se zobrazovala hned na začátku, kdy po jejím přijetí už nebylo nutné další údaje explicitně nastavovat a služba tím byla připravena k vytvoření.

	Id	DeviceID	Time	Temperature	Level	F...	F...	D...	D...	Mix...	Heat...
1	1882	Simulator_...	2025-05-17 11:16:44.173	24,62384	0,1869213	1	0	1	0	0	0
2	1881	Simulator_...	2025-05-17 11:16:33.907	24,84809	0,1793258	1	0	1	0	0	0
3	1880	Simulator_...	2025-05-17 11:16:23.303	25,08319	0,1940104	1	0	1	0	0	0
4	1879	Simulator_...	2025-05-17 11:16:12.713	24,93128	0,2194011	1	0	1	0	0	0
5	1878	Simulator_...	2025-05-17 11:16:02.407	25,34361	0,1770833	1	0	1	0	0	0
6	1877	Simulator_...	2025-05-17 11:15:51.770	25,99465	0,1986401	1	0	1	0	0	0
7	1876	Simulator_...	2025-05-17 11:15:41.270	25,61126	0,1770833	1	0	1	0	0	0
8	1875	Simulator_...	2025-05-17 11:15:30.883	26,68548	0,2168692	1	0	1	0	0	0
9	1874	Simulator_...	2025-05-17 11:15:22.147	27,6331	0,1786024	1	0	1	0	0	0
10	1873	Simulator_...	2025-05-17 11:15:09.723	29,10518	0,2005932	1	0	1	0	0	0
11	1872	Simulator_...	2025-05-17 11:14:59.600	30,85576	0,1841725	1	0	1	0	0	0
12	1871	Simulator_...	2025-05-17 11:14:48.850	34,25926	0,2247541	1	0	1	0	0	0
13	1870	Simulator_...	2025-05-17 11:14:38.293	38,82379	0,2038484	1	0	1	0	0	0

Obr. 4.13: Ukázka naplněné databáze

V sekci Přehled je volba Nastavit bránu firewall serveru, kde se přidává IP adresa zařízení, které se může do databáze přihlásit. Do databáze se přihlašuje v sekci Editor dotazů (Preview), kde je možné spouštět SQL dotazy nad tabulkami databáze. Aby se dala databáze naplnit daty, musí se v ní vytvořit tabulka, bez které je databáze pouze prázdný kontejner bez struktury. Pomocí příkazu CREATE Table byla vytvořena tabulka, která odpovídala struktuře příchozích dat ze simulátoru.

Databáze se plnila daty pomocí Azure funkce, která byla doplněna o připojovací řetězec. Připojovací řetězce se nacházejí v sekci Nastavení, kdy jsou rozděleny podle typu ověřování. Do Azure funkce byla vložena statická proměnná a připojovací řetězec se jménem a heslem byl opět vložen do IoT Hubu do sekce Nastavení, respektive podsekce Proměnné prostředí, jak bylo popsáno v tvorbě Azure funkce. Z bezpečnostních důvodů není heslo součástí kopírovaného připojovacího řetězce a musí se doplnit ručně. Azure funkce následně tento připojovací řetězec využívá pro spojení s databází, do které zapisuje data ze zprávy z IoT Hubu.

Pro pohodlnější práci s databází byl využit software Azure Data Studio, který nabízí přehlednější prostředí než je v Azure portálu. Připojení do databáze je možné dvěma způsoby - buď opět přes připojovací řetězec, nebo ručním vyplněním údajů ohledně serveru, jména databáze a způsobu ověření. Po úspěšném připojení se zobrazí nastavitelné okno pro spouštění dotazů a zobrazení výsledků.

4.3.3 Zpětná komunikace

Princip digitálního dvojčete zahrnuje nejen zobrazení stavu fyzického zařízení, ale také jeho automatickou odezvu. Ta je realizována opět pomocí Azure funkce, která přečte data z digitálního dvojčete a odešle je do TwinCATu. Odtud je možné pak na zprávu zareagovat a přímo ovládat nebo parametrizovat simulátor.

Realizace odezvy byla přidána do aktuální Azure funkce. Spočívá v přečtení aktuálních hodnot, které jsou uloženy do proměnných datového objektu a převedeny do JSON zprávy. Výsledný řetězec je poté odeslán jako zpráva cloud-to-device do zařízení prostřednictvím IoT Hubu. Zařízení zprávu zpracuje a uloží do fronty. Ve frontě může být maximálně 50 zpráv, další zprávy pak způsobí chybu. Zprávy jsou obvykle po 24 hodinách smazány a již není možné je vyzvednout. V TwinCATu byl implementován jednoduchý kód na příjem zprávy přes protokol MQTT. Klíčové je správné nastavení tématu, které zajistí odběr všech zpráv, které byly do zařízení poslány. Přijatá zpráva je pak ve formátu JSON uložena v proměnné sPayloadRcv.

4.4 Metody realizace

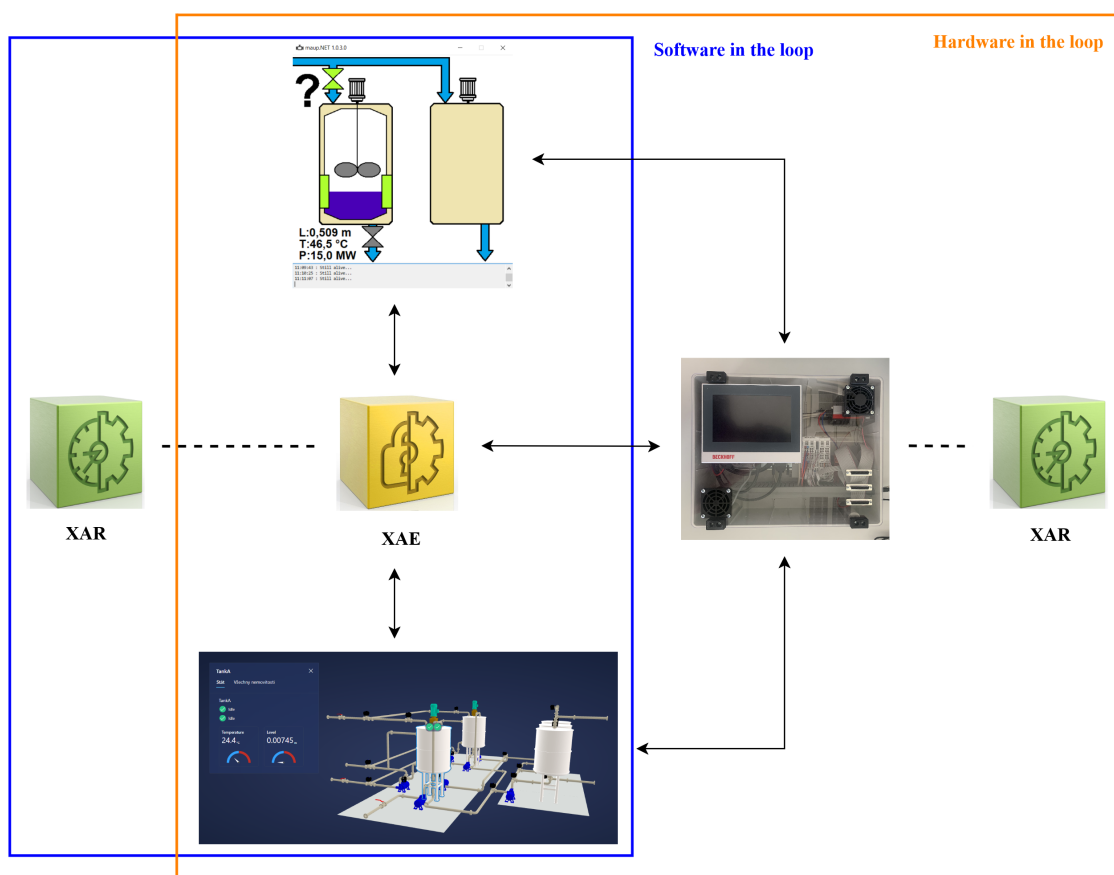
V rámci vývoje a testování moderních průmyslových systémů je důležitou součástí software, který je potřeba před samotným nasazením do provozu vždy pořádně otestovat. K tomu se využívají metody Software in the Loop (SiL) a Hardware in the Loop (HiL). Jedná se o různé typy propojení, kdy se v případě SiL využívá počítač bez reálného hardwaru, oproti tomu HiL využívá fyzický řídicí systém.

Software in the loop

Metoda SiL spočívá v tom, že řídicí algoritmus běží v softwarovém prostředí, kde je propojen s modelem simulujícím fyzikální proces. Používá se pro usnadnění ladění a vývoje celého systému, protože testování probíhá před nasazením do reálného hardwaru. V pozdější fázi může běžet paralelně s fyzickým systémem, čímž vznikne jak reálný výrobek, tak jeho digitální kopie.

Hardware in the loop

Na rozdíl od metody SiL je zde model řízen řídicím algoritmem přímo z reálného PLC. V tomto případě nemusí být zařízení ještě fyzicky realizováno. Lze tak otestovat interakci s reálným hardwarem, která může přinést v případě poškození zbytečné náklady navíc. Slouží tak pro ověření robustnosti a připravenosti na reálný provoz. V některých případech může taktéž sloužit pro trénink obsluhy zařízení. [34]



Obr. 4.14: Propojení procesu

4.4.1 Využití obou metod

Na obrázku 4.14 je zobrazena architektura Pasterizační jednotky využívající obě výše uvedené metody. Modře označená část představuje využití metody SiL, kdy probíhá simulace řídicího algoritmu v prostředí TwinCAT. Obě hlavní komponenty TwinCATu běží na vývojovém počítači a komunikace se simulátorem probíhá prostřednictvím vnitřních smyček Ethernetu. Přenos dat do cloudu, respektive služby Azure Digital Twins s využitím Azure IoT Hubu, vychází rovněž z vývojového počítače.

Naopak v případě metody HiL, která je označena oranžovou barvou, je řídicí algoritmus nasazen na průmyslovém panelu s prostředím TwinCAT XAR. Komunikace se simulátorem v počítači probíhá pomocí protokolu Modbus TCP, kdy fyzickou vrstvu tvoří Ethernet kabel. Komunikace s cloudem je realizována také přímo z průmyslového panelu. Tato struktura umožňuje přechod z vývojového prostředí k provoznímu prostředí v souladu s principy Průmyslu 4.0. V pravé dolní části je umístěna legenda, která znázorňuje jednotlivá fyzická propojení včetně použitých komunikačních protokolů.

Je potřeba zdůraznit, že metody HiL a SiL nejsou v tomto případě využity pro simulaci modelu fyzického zařízení, ale pro ověření spolehlivé komunikace mezi jednotlivými částmi architektury. V obou případech je reálný proces nahrazen simulátorem Pasterizační jednotky. Cílem je tedy demonstrovat, že digitální dvojče v 3D Scenes správně přijímá a vizualizuje data jak z vývojového počítače, tak z průmyslového panelu. Zároveň se ověřuje, že komunikace je obousměrná a stabilní.

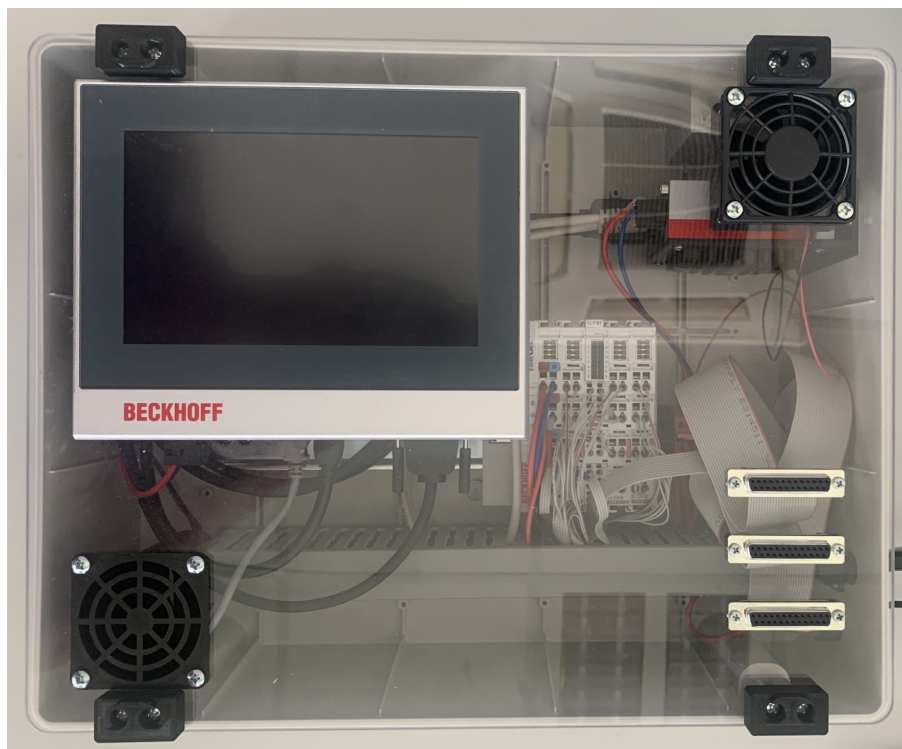
4.5 Fyzická vrstva

Tato kapitola se zabývá propojením fyzického zařízení s digitálním dvojčetem. Popisuje použitý hardware, postup propojení a konfiguraci síťových parametrů pro stabilní spojení. Zaměřuje se také na způsob nastavení komunikace prostřednictvím protokolu Modbus TCP, který je využit pro komunikaci se simulátorem.

4.5.1 Propojení s fyzickým panelem

Pro realizaci propojení fyzického zařízení s digitálním dvojčetem byl použit průmyslový řídicí systém od společnosti Beckhoff. Tento systém je umístěn v plastovém boxu, kdy se v horní části nachází dotykový HMI panel, který běží na operačním systému Windows 10. Uvnitř boxu je robustní základní deska Beckhoff CB6273, která slouží jako hlavní výpočetní jednotka tohoto systému. Na víku boxu jsou umístěny dva ventilátory pro chlazení celého systému. Uvnitř boxu jsou DIN lišty s EtherCAT

moduly pro zpracování vstupních a výstupních signálů. Součástí systému je také Ethernet adaptér zajišťující síťovou komunikaci. Na boční straně boxu se nacházejí tři konektory pro Ethernet, napájení a USB. Vedle konektorů se nachází také tlačítko k zapnutí panelu. Součástí panelu je nainstalované runtime prostředí TwinCAT XAR. Celé zařízení je určeno pro výuku a testování automatizačních aplikací v prostředí TwinCAT.



Obr. 4.15: Průmyslový panel Beckhoff

Do této chvíle to byla situace, kdy byl jak PLC program, tak i simulátor spuštěn na jednom počítači. Program běžel lokálně v prostředí TwinCAT softwarového PLC a komunikace se simulátorem přes Modbus tak probíhala v rámci jednoho zařízení. Při nahrávání programu do fyzického zařízení, respektive do Beckhoff průmyslového panelu, je nutné, aby počítač i panel byly připojeny do stejné sítě. Po fyzickém připojení Ethernet kabely do sítě bylo zjištěno číslo sítě, číslo podsítě, maska podsítě, výchozí brána sítě a DNS server.

Tyto parametry byly fixně zapsány, jak do průmyslového panelu, tak do počítače. Toto nastavení se provádí v počítači v položce Protokol IP verze 4 (TCP/IPv4) ve vlastnostech Ethernetu. Na obrázku 4.16 jsou zobrazeny parametry, bez nichž komunikace nebude fungovat. Oběma zařízením byla přiřazena jedinečná IP adresa v daném rozsahu. Test správného nastavení parametrů byl proveden pomocí příkazu ping z příkazového řádku, kde bylo ověřeno, že se obě zařízení v síti navzájem vidí.

Obecné

Podporuje-li síť automatickou konfiguraci IP, je možné získat nastavení protokolu IP automaticky. V opačném případě vám správné nastavení poradí správce sítě.

☐ Získat IP adresu ze serveru DHCP automaticky

☒ Použít následující IP adresu:

IP adresa:

10 . 0 . 140 . 77

Maska podsítě:

255 . 255 . 0 . 0

Výchozí brána:

10 . 0 . 0 . 1

☐ Získat adresu serveru DNS automaticky

☒ Použít následující adresy serverů DNS:

Upřednostňovaný server DNS:

8 . 8 . 8 . 8

Alternativní server DNS:

8 . 8 . 8 . 4

☐ Při ukončení ověřit platnost nastavení

Upřesnit...

OK

Zrušit

Obr. 4.16: Nastavení síťových parametrů v počítači

Nastavení TwinCATu

V horní liště je zobrazen název zařízení, které je momentálně připojeno. Dosud tam byl zobrazen název <Local> pro softwarové PLC. Pro připojení ke konkrétnímu zařízení se kliknutím na název zobrazí volba Choose Target System. V ní se hledáním zařízení přes Ethernet a následně volby Broadcast search zobrazí všechna dostupná zařízení v síti s aktivním XAR. Po výběru cílového zařízení dle nastavené IP adresy se přidá tzv. route, tedy cesta pro komunikaci. Pro ni je vyžadováno nastavení zabezpečení, kdy se zvolí normální zabezpečení a přihlášení přes defaultní uživatelské jméno Administrator a heslo 1. Jakmile je cesta nastavena, zobrazí se u připojeného zařízení v kolonce Connected křížek, indikující správně nastavenou cestu. Aktivací volby Activate Configuration se spustí program na cílovém zařízení.

Pro úspěšné odesílání zpráv na IoT Hub bylo nezbytné nahrát kořenový certifikát do průmyslového panelu. Jak již bylo zmíněno, panel využívá operační systém Windows 10, tudíž existuje možnost stáhnout certifikát stejným způsobem, jako do počítače. Jednodušší variantou je však přenést na panel již vytvořený certifikát z počítače prostřednictvím USB flash disku a umístit jej například na plochu. V kódu se následně nastavuje přímá cesta k certifikátu. V kódu je nastavena cesta k připojenému USB flash disku.

4.5.2 Komunikace přes Modbus

Pro funkční Modbus komunikaci je v programovém bloku MB_Connector nutné změnit IP adresu. V případě softwarového PLC byla adresa nastavena na hodnotu '127.0.0.1', představující tzv. loopback adresu, jelikož TwinCAT komunikoval s Modbus serverem v rámci jednoho zařízení. Pokud dochází ke komunikaci s fyzickým zařízením, jako je tento případ, byla nastavena IP adresa počítače, tj. 10.0.140.77.

Dalším krokem bylo zajistit přístupnost portu 503 nastavením pravidla ve firewallu počítače v programu Windows Defender s pokročilým zabezpečením. Vzhledem k problémům s Modbus komunikací bylo dočasně provedeno úplné vypnutí firewallu. Vypnutí firewallu vzniklé problémy vyřešilo, avšak z hlediska bezpečnosti představuje riziko neoprávněného připojení k počítači zvenčí nebo staženou aplikací. Díky těmto krokům pak bylo dosaženo propojení fyzického zařízení, komunikujícího přes Modbus komunikaci s technologickým procesem, respektive simulátorem, s digitálním dvojčetem.

Závěr

Cílem této diplomové práce bylo propojit koncept Průmyslu 4.0 s cloudovou platformou Microsoft Azure prostřednictvím vytvoření digitálního dvojčete.

Teoretická část byla zaměřena na seznámení se s konceptem Průmyslu 4.0, přičemž hlavní pozornost byla věnována technologii digitálního dvojčete. Byly popsány a vysvětleny jednotlivé fáze jeho vývoje. Následně byla stručně představena platforma Microsoft Azure a služby, které byly v práci využity. Jedná se o rozsáhlou cloudovou platformu se spoustou nástrojů a funkcionalit, jejichž popis by přesahoval rámec této práce. V závěru teoretické části byl popsán také princip funkce technologického procesu Pasterizační jednotky.

V rámci realizace bylo plánováno využít technickou dokumentaci od společnosti Microsoft. Ukázalo se však, že oficiální návody nebyly aktuální nebo neodpovídaly požadavkům řešení, a proto bylo nezbytné jednotlivé kroky nastudovat od samotného začátku. Tento způsob si vyžádal více času, než se původně předpokládalo. Důležitým krokem byla implementace MQTT komunikace pro přenos zpráv z TwinCATu do IoT Hubu. Tyto zprávy byly následně zpracovány pomocí Azure funkce, jejíž ladění bylo zásadní pro celou práci. Výsledkem pak bylo úspěšné zobrazení dat v grafu digitálních dvojčat služby Azure Digital Twins.

Další část byla soustředěna na vytvoření 3D modelu Pasterizační jednotky, včetně technologie čištění. Tvorba modelu byla provedena v Siemens NX, přičemž byla poměrně časově náročná. Tento model byl pak následně importován do nástroje 3D Scenes, kde bylo nakonfigurováno grafické rozhraní pro vizualizaci. Díky tomu, že se nástroj 3D Scenes nachází ve službě Azure Digital Twins, bylo propojení s daty ze simulátoru jednoduché. Na závěr byl celý proces integrován do fyzického zařízení, respektive do průmyslového panelu. Došlo tak k propojení digitálního dvojčete s fyzickým zařízením, ve kterém byl implementován řídicí algoritmus. Procesní data ze simulátoru byla zároveň ukládána do SQL databáze k pozdějšímu využití.

Navazující prací může být analýza těchto dat pro realizaci adaptivního regulátoru. Řídicí algoritmus v TwinCATu obsahuje PID regulátor. Na základě odezvy na jednotkový skok a následným využitím metod pro návrh regulátoru by mohl být navržen regulátor, který bude aktualizovat své složky za běhu procesu.

Literatura

- [1] Andrea Cejnarová. Od 1. průmyslové revoluce ke 4. [online]. [cit. 2025-05-20]. URL: https://www.technickytydenik.cz/rubriky/ekonomika-byznys/od-1-prumyslove-revoluce-ke-4_31001.html.
- [2] F. Zezulka, J. Jirsa, L. Venkrbec, P. Marcon, T. Benesl, V. Kaczmarczyk, J. Arm, and Z. Bradac. The ideas of industry 4.0: Seven years after. *IFAC-PapersOnLine*, 55(4):145–150, 2022. [cit. 17th IFAC Conference on Programmable Devices and Embedded Systems PDES 2022 — Sarajevo, Bosnia and Herzegovina, 17-19 May 2022]. URL: <https://www.sciencedirect.com/science/article/pii/S2405896322003391>, doi:10.1016/j.ifacol.2022.06.024.
- [3] Núbia Gabriela Pereira Carvalho and Edson Walimir Cazarini. Industry 4.0 - what is it? In *Industry 4.0 - Current Status and Future Trends*, pages 3–11. 2020. [cit. 2025-01-05]. URL: <http://dx.doi.org/10.5772/intechopen.90068>.
- [4] Václav Kaczmarczyk. *Průmysl 4.0 a digitální dvojče*. Habilitační práce, Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, Brno, 2023.
- [5] Paula Morella, M Lambán, Jesús Royo, J Sánchez, and O Muñoz. Cyber physical systems implementation to develop a smart manufacturing. *IOP Conference Series: Materials Science and Engineering*, 1193:012114, 10 2021. doi:10.1088/1757-899X/1193/1/012114.
- [6] Elisa Yumi Nakagawa, Pablo Oliveira Antonino, Frank Schnicke, Rafael Capilla, Thomas Kuhn, and Peter Liggesmeyer. Industry 4.0 reference architectures: State of the art and future trends. *Computers & Industrial Engineering*, 156:107241, 2021. URL: <https://www.sciencedirect.com/science/article/pii/S0360835221001455>, doi:10.1016/j.cie.2021.107241.
- [7] Co je iaas? [online]. [cit. 2025-01-06]. URL: <https://azure.microsoft.com/cs-cz/resources/cloud-computing-dictionary/what-is-iaas>.
- [8] Obrázek 8: Pillars of industry 4.0. [online]. [cit. 2025-01-06]. URL: https://www.researchgate.net/figure/Pillars-of-Industry-40-Source-adapted-from-IIES-2017\protect\discretionary{\char\hyphenchar\font}{}{}Liao-et-al-2017s-study_fig2_333578151.
- [9] Jesús Hamilton Ortiz, William Gutierrez Marroquin, and Leonardo Zambrano Cifuentes. Industry 4.0: Current status and future trends. In *Industry 4.0 -*

- Current Status and Future Trends*, pages 13–28. 2020. [cit. 2025-01-05]. URL: <http://dx.doi.org/10.5772/intechopen.90396>.
- [10] Pieter van Schalkwyk and Shyam Varan Nath. *Building Industrial Digital Twins*. Packt Publishing, Birmingham, 2021.
 - [11] Mohsen Attaran and Bilge Gokhan Celik. Digital twin. *Decision Analytics Journal*, 6, 2023. [cit. 2025-01-06]. URL: <https://linkinghub.elsevier.com/retrieve/pii/S277266222300005X>, doi:10.1016/j.dajour.2023.100165.
 - [12] Dr. Michael Grieves. [figure 1: Most common model of the dm, ds, dt taxonomy]. [online], 2023. [cit. 2025-01-06]. URL: https://www.researchgate.net/figure/fig1_369830792.
 - [13] Dr. Michael Grieves. Digital model, digital shadow, and digital twin. *ResearchGate*, pages 1–3, 2023. [cit. 2025-01-06]. URL: https://www.researchgate.net/publication/369830792_Digital_Model_Digital_Shadow_Digital_Twin.
 - [14] Mohsen Soori, Behrooz Arezoo, and Roza Dastres. Digital twin for smart manufacturing, a review. *Sustainable Manufacturing and Service Economics*, 2, 2023. [cit. 2025-01-06]. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2667344423000099>, doi:10.1016/j.smse.2023.100017.
 - [15] Co je mqtt a k čemu slouží ve iiot? popis protokolu mqtt. [online]. [cit. 2025-01-06]. URL: <https://ipc2u.cz/blogs/news/mqtt-protokol>.
 - [16] Eclipse mosquitto™. [online]. [cit. 2025-01-06]. URL: <https://mosquitto.org/>.
 - [17] Podrobný popis protokolu modbus tcp s příklady příkazů. [online]. [cit. 2025-01-06]. URL: <https://ipc2u.cz/blogs/news/podrobny-popis-protokolu-modbus-tcp-s-priklady-prikazu>.
 - [18] Seznamte se s azure. [online]. [cit. 2025-01-06]. URL: <https://azure.microsoft.com/cs-cz/explore>.
 - [19] Co je služba azure digital twins? [online]. [cit. 2025-01-06]. URL: <https://learn.microsoft.com/cs-cz/azure/digital-twins/overview>.
 - [20] Co je azure iot hub? [online]. [cit. 2025-01-06]. URL: <https://learn.microsoft.com/cs-cz/azure/iot-hub/iot-concepts-and-iot-hub>.
 - [21] Instalace a použití azure iot exploreru. [online]. [cit. 2025-01-06]. URL: <https://learn.microsoft.com/cs-cz/azure/iot/howto-use-iot-explorer>.

- [22] Co je azure data studio? [online], 2025. [cit. 2025-05-20]. URL: <https://learn.microsoft.com/cs-cz/azure-data-studio/what-is-azure-data-studio>.
- [23] Michal Husák. *Návrh a realizace laboratorní úlohy na téma "Standard ISA-S88*. Diplomová práce, Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, Brno, 2018.
- [24] Václav Kaczmarczyk. Mpc aup. [online], 2024. [cit. 2025-05-20]. URL: https://docs.google.com/document/d/1-GaU5kCZ1GRN1NvInL0n1PCSKo1_SiEk/edit?rtf=true&sd=true&tab=t.0.
- [25] Philosophy. [online]. [cit. 2025-05-20]. URL: https://infosys.beckhoff.com/english.php?content=../content/1033/tf6720_tc3_iot_data_agent/3528168587.html&id=.
- [26] Ads-communication. [online]. [cit. 2025-05-20]. URL: https://infosys.beckhoff.com/english.php?content=../content/1033/cx8190_hw/5091854987.html&id=.
- [27] [microsoft azure iot hub and twincat iot]. [online]. [cit. 2025-01-06]. URL: https://infosys.beckhoff.com/english.php?content=../content/1033/tf6720_tc3_iot_data_agent/3528168587.html&id=.
- [28] Beckhoff#let's send data to azure iot hub with the twincat tf6701. [online], 2016. [cit. 2025-05-20]. URL: <http://soup01.com/en/2024/01/24/beckhofflets-send-data-to-azure-iot-hub-with-the-twincat-tf6701/>.
- [29] Iotmqttsampleazureiothub. [online]. [cit. 2025-01-06]. URL: https://infosys.beckhoff.com/english.php?content=../content/1033/tf6701_tc3_iot_communication_mqtt/3528172299.html&id=.
- [30] Iotmqttsampleusingqueue. [online]. [cit. 2025-01-26]. URL: https://infosys.beckhoff.com/english.php?content=../content/1033/tf6701_tc3_iot_communication_mqtt/3391755403.html&id=.
- [31] Kurz: Vytvoření a nahrání certifikátů pro testování. [online]. [cit. 2025-01-06]. URL: <https://learn.microsoft.com/cs-cz/azure/iot-hub/tutorial-x509-test-certs?tabs=windows>.
- [32] Nx (program). [online], 2001-. [cit. 2025-05-20]. URL: [https://cs.wikipedia.org/wiki/NX_\(program\)](https://cs.wikipedia.org/wiki/NX_(program)).
- [33] Co je azure event grid? [online], 2025. [cit. 2025-05-20]. URL: <https://learn.microsoft.com/cs-cz/azure/event-grid/overview>.

- [34] Michal Husák. *Virtuální dvojče pro testbed Průmyslu 4.0*. Bakalářská práce, Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, Brno, 2020.

A Obsah elektronické přílohy

Elektronická příloha je rozdělena do 3 složek podle využitých softwarů. Ve složce Siemens NX jsou obě hlavní sestavy, ze kterých je vytvořen finální model. Obsahuje tento model také ve formátu .glb pro import do 3D Scenes. Mimo to obsahuje také jednotlivé díly sestav. Sestavy byly vytvořeny ve verzi Siemens NX 2306.7000.

Ve složce TwinCAT je hlavní projekt MCPAUP_Tanky. Projekt byl vytvořen v TwinCAT 3. Ve složce Visual_studio je složka obsahující modely zařízení v jazyce DTDL a složku s projektem. Projekt byl vytvořen v Microsoft Visual Studiu 2022.

```
/ .....kořenový adresář
├── dp_dvorak_tomas ..... hlavní složka
│   ├── Siemens NX
│   │   ├── simulator_tanku ..... sestava hlavního tanku
│   │   ├── cistení ..... sestava technologie čištění
│   │   ├── final_model_pasterizacni_jednotka ..... finální model
│   │   └── pasterizace ..... finální model ve formátu .glb
│   ├── TwinCAT
│   │   └── MCPAUP_Tanky ..... hlavní projekt
│   └── Visual_studio
│       ├── dp_HubToAzure
│       │   └── dp_HubToAzure ..... hlavní projekt
│       └── modely
```