

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA ELEKTROMECHANIKY A VÝKONOVÉ ELEKTRONIKY

DIPLOMOVÁ PRÁCE

Návrh řídicího systému pro technologický proces

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: 2018/2019

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Marek SLÁMA**
Osobní číslo: **E17N0028P**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Komerční elektrotechnika**
Název tématu: **Návrh řídicího systému pro technologický proces**
Zadávací katedra: **Katedra technologií a měření**

Z á s a d y p r o v y p r a c o v á n í :

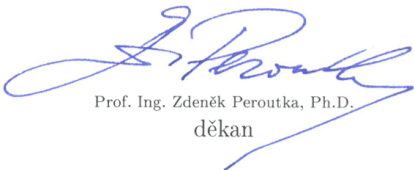
Cílem diplomové práce je navrhnout a realizovat řídicí systém pro řízení technologického procesu. Diplomová práce bude realizována ve spolupráci se soukromou společností, která poskytne konkrétní zadání technologického procesu a potřebný HW a SW pro zpracování diplomové práce.

1. Charakterizujte technologický proces a klíčové komponenty technologie, jejíž řízení budete realizovat. Zpracujte technologické zadání do formy konkrétních funkčních popisů. Definujte základní požadavky na systém kontroly a řízení.
2. Navrhněte konfiguraci řídicího systému.
3. Zpracujte na základě technologických funkčních popisů funkční popis pro systém kontroly a řízení včetně systému značení, tabulky vstupů a výstupů a algoritmů/sekvencí.
4. Implementujte navržené algoritmy řízení do DCS. Navrhněte a realizujte vizualizaci řízení technologického procesu pro HSI.
5. Otestujte správnou implementaci funkčních popisů a vazby na HSI.

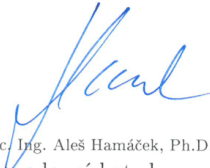
Rozsah grafických prací: **podle doporučení vedoucího**
Rozsah kvalifikační práce: **40 - 60 stran**
Forma zpracování diplomové práce: **tištěná/elektronická**
Seznam odborné literatury:

- 1. ZEŽULKA, František. Prostředky průmyslové automatizace. Brno: VUTIUM, 2004. ISBN 80-214-2610-1.**

Vedoucí diplomové práce: **Ing. Martin Sirový, Ph.D.**
Regionální inovační centrum elektrotechniky
Konzultant diplomové práce: **Ing. Petr Froněk**
Datum zadání diplomové práce: **5. října 2018**
Termín odevzdání diplomové práce: **30. května 2019**


Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan




Doc. Ing. Aleš Hamáček, Ph.D.
vedoucí katedry

V Plzni dne 5. října 2018

Abstrakt

Předložená práce je věnována procesu tvorby průmyslové automatizace, kterou prakticky demonstruje na uceleném řešení vybrané části technologického celku. Práce definuje základní pojmy užívané v automatizaci, popisuje jednotlivé úrovně řídicího systému a způsobu komunikace v průmyslové automatizaci. Součástí práce je jednak komplexní popis procesu návrhu průmyslové automatizace pro vybranou část technologického celku, včetně návrhu uživatelského rozhraní, jednak vlastní implementace a základní úroveň testování v simulačním prostředí řídicího systému. Technologický celek je součástí čistírny odpadních vod. Tato práce vznikla ve spolupráci se společností ABB, pro řídicí systém byla využita platforma 800xA. Čtenář bude seznámen s tvorbou programu pro programovatelné kontroléry za pomoci softwaru Control Builder od společnosti ABB. Čtenář by měl ale být schopen porozumět i systému jako celku, ve kterém je program Control Builderu jen dílčí částí. Postup programování je podrobně demonstrován na vybraných částech systému a je napsán takovým způsobem, aby byla dodržena časová a logická posloupnost vytváření programu. Celý program je pak přílohou na přiloženém nosiči.

Klíčová slova

AC 800M, ABB 800xA, PLC, Control builder, HSI, DCS, HMI řídicí systém, programovatelný logický automat

Abstract

This diploma work is focused on process creation of industrial automation, which is demonstrated on the solution of particular part of technology unit. The work defines basic terms used in automation, describes singular levels of control system and the industrial automation communication protocols. The industrial automation design for a particular part of technology unit is described fully, including the user interface design, and own implementation and the elementary level of test in simulated environment of control system is added as well. The technology unit is a part of wastewater treatment plant. This diploma work was written in cooperation with the company ABB, and the platform 800xA was used for the control system. The reader will be acquainted with the output of program for programmable logic controllers using the software Control Builder by the company ABB. The reader should be able to understand the system as the whole, in which the program Control Builder presents only a partial section. The process of programming is demonstrated completely on particular parts of system in order to comply time and logic sequence of the program's creation. The whole program is available on the memory card as well.

Key words

AC 800M, ABB 800xA, programmable logic controller, Control builder, HMI, Human System Interface, HSI, PLC, Control System, DCS

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

.....

podpis

V Plzni dne 29.5.2019

Marek Sláma

.....

Poděkování

Tímto bych rád poděkoval vedoucímu diplomové práce, kterým byl pan Ing. Martin SIROVÝ, Ph.D. za čas strávený konzultacemi a také za připomínky, doporučení a metodické vedení práce. Dále bych chtěl poděkovat panu Ing. Petru Froňkovi za cenné profesionální rady a za jeho čas věnovaný mému zaučení. Děkuji také vedoucímu plzeňské pobočky společnosti ABB panu Ing. Jirkovi Chvalovi za to, že mi umožnil tuto práci vypracovat.

Obsah

OBSAH.....	8
ÚVOD.....	10
SEZNAM SYMBOLŮ A ZKRATEK.....	11
1 PROBLEMATIKA AUTOMATICKÉHO ŘÍZENÍ.....	12
1.1 PROGRAMOVATELNÝ LOGICKÝ AUTOMAT.....	12
2 SYSTÉMY PRŮMYSLOVÉ AUTOMATIZACE	14
2.1 HIERARCHIE SYSTÉMU PRŮMYSLOVÉ AUTOMATIZACE.....	16
2.2 CENTRALIZOVANÉ A DISTRIBUOVANÉ SYSTÉMY.....	17
2.3 KOMUNIKACE V PRŮMYSLOVÉ AUTOMATIZACI	18
2.3.1 Profibus.....	18
3 NÁVRH SYSTÉMU ŘÍZENÍ REÁLNÉ ÚPRAVNY VOD	20
3.1 PROCES NÁVRHU ŘÍDICÍHO SYSTÉMU.....	20
3.2 POPIS TECHNOLOGICKÉHO CELKU	21
3.2.1 Oddělovač před čistírnou	21
3.2.2 Lapák štěrku.....	22
3.2.3 Česle.....	22
3.2.4 Lapák písku.....	22
3.2.5 Lapáky tuků a olejů.....	22
3.2.6 Usazovací nádrže.....	22
3.2.7 Biologické čištění.....	23
3.3 PIPING AND INSTRUMENTATION DIAGRAM	23
3.3.1 Systém značení.....	23
3.3.2 Výčet objektů.....	23
3.4 ZÁKLADNÍ POŽADAVKY NA SYSTÉM KONTROLY A ŘÍZENÍ	25
3.5 HIERARCHIE ŘÍDICÍHO SYSTÉMU.....	25
3.6 TECHNOLOGICKÝ POPIS ZPRACOVÁVANÉHO CELKU.....	26
3.6.1 Lapák písku.....	26
3.6.2 Škrabák tuku	27
3.7 FUNKČNÍ POPIS.....	27
3.8 KONFIGURACE SYSTÉMU.....	31
3.8.1 Alokace signálů.....	32
3.9 VYBRANÝ SYSTÉM ŘÍZENÍ.....	32
3.9.1 Systém 800xA.....	33
3.9.2 Koncept objektu a aspektu.....	33
3.9.3 Struktury Systému.....	34
4 IMPLEMENTACE NAVRŽENÉHO ŘEŠENÍ DO SYSTÉMU.....	36
4.1 VYTVOŘENÍ NOVÉHO PROJEKTU	36
4.2 PŘIDÁVÁNÍ KNIHOVEN DO PROJEKTU	36
4.3 NASTAVENÍ HARDWARU	37
4.4 PŘIDÁVÁNÍ HW MODULŮ.....	39
4.5 POSTUP PROGRAMOVÁNÍ.....	39
4.5.1 Alokace signálů.....	41
4.5.2 Výběr vhodné metody programování.....	44
4.5.3 Práce v diagramu funkčních bloků.....	44
4.5.4 Strukturovaný datový typ	48
4.5.5 Implementace PROFIBUS komunikace.....	49

4.5.6	Ovládání motorového bloku.....	54
4.5.7	Možnosti programu Control Builder	56
5	TVORBA GRAFICKÉHO ROZHRANÍ.....	57
5.1	GRAFICKÉ OBJEKTY.....	57
5.2	FILIZOFIE BAREV	60
5.3	GRAFICKÉ ROZHRANÍ LAPÁKU PÍSKU	61
6	TESTOVÁNÍ.....	63
7	ZÁVĚR.....	67
	SEZNAM POUŽITÉ LITERATURY.....	69
	SEZNAM OBRÁZKŮ.....	73
	SEZNAM TABULEK	74
	SEZNAM PŘÍLOH.....	75
8	PŘÍLOHY.....	76

Úvod

Jedním z trendů dnešní doby je automatizace. Stále více jsme odkázáni na technologie a automatizaci. Důvodem zavádění automatizace je náročnost řízení složitých technologií, vysoká spolehlivost automatizovaných systémů, úspora financí, lidské síly a času. Nahrazení lidské obsluhy nese kromě snížení námahy jednu velkou výhodu – Odstranění tzv. lidského faktoru a tím snížení chyb. V případě, že se bude zvětšovat počet zařízení kolem nás, budeme nuceni tato zařízení naučit pracovat s minimem lidské činnosti. Automatizace také nabízí větší míru přesnosti regulace ve srovnání s člověkem ovládanými systémy. Pro řízení průmyslových technologií jsou hojně využívány programovatelné logické kontroléry – PLC. Mozkem pro ovládání technologie v této práci je právě PLC. Nedílnou součástí automatizace je i komunikace mezi jednotlivými zařízeními a s tím související komunikační protokoly. V práci jsou obsaženy používané typy komunikačních protokolů a dále také popsán vybraný způsob komunikace.

Práce si klade za cíl objasnit princip návrhu řídicího systému pro technologický proces, za pomoci PLC, postup jeho konfigurace, metody programování PLC, způsob implementace a tvorby rozhraní HSI (Human System Interface). V práci je obsažen postup realizace těchto kroků na vybraném technologickém celku. Práce je díky realizaci skutečného řízení technologického celku napsána tak, aby čtenářovi ve svých kapitolách nabídla jak teoretické, tak i praktické znalosti. Vytváření programu je popsáno tak, aby se podobné kroky neopakovaly a celý program je pak přílohou na samostatném nosiči. Pozornost je věnována popisu klíčových komponentů technologie a konkrétním funkčním popisům. Práce obsahuje návrh konfigurace řídicího systému vybraného technologického celku. Vybraný technologický celek je součástí úpravny vod. Součástí práce je i popis technologie. Podnětem pro vznik práce byla potřeba vytvořit ucelený dokument popisující jednotlivé aspekty návrhu řídicího systému. Součástí práce je i popis testování vytvořeného řízení včetně grafického rozhraní.

Čtenář této práce bude seznámen se systémem společnosti ABB. Použitý systém nese označení 800xA a je nadřazeným systémem, který zahrnuje komunikaci, kontroléry, vstupní a výstupní moduly, síť, zdroje a jsou v něm implementovány programy i operátorské panely. Složitá architektura tohoto systému je v práci vysvětlena. Čtenář bude seznámen také s konkrétním hardwarem, který je využíván v průmyslové automatizaci a měl by pochopit jeho význam v řídicí struktuře.

Seznam symbolů a zkratek

PLC	Programmable Logic Controller (Programovatelný logický kontrolér)
CPU	Central Processor Unit (Centrální Procesorová Jednotka)
HMI	Human Machine Interface (Rozhraní člověk stroj)
HSI	Human System Interface (Rozhraní mezi člověkem a systémem)
DCS	Distribution Control System (Distribuovaný systém řízení)
FPGA	Field Programmable Gate Array
SRAM	Static Random Access Memory
RAM	Random-Access Memory
ROM	Read-Only Memory
I/O	Input / Output
HW	Hardware
FBD	Function Block Diagram (Diagram funkčních bloků)
PROFIBUS	Process Field Bus
DP	Decentralized Periphery
GSD	General Station Description
FAT	Factory Acceptance Test
SAT	Site Acceptance Test

1 Problematika automatického řízení

Za automatické řízení je považováno samočinné působení definovanými instrukcemi podle vloženého programu na řízený celek s účelem dosažení vytyčeného cíle [1]. Do konce 70. let ve světě automatizace dominovala relé technologie [2]. V reálném provozu bylo zapotřebí velké množství těchto relátek i k ovládní poměrně jednoduché logiky a u složitějších provozů počet relátek strmě rostl. Rostla tak i složitost odstraňování případných poruch a zavádění systému do provozu. U složitějších systémů zabíraly takové kontakty také velkou spoustu místa. Další nevýhodou byla skutečnost, že se jednalo o mechanický kontakt vyžadující přísný plán údržby [3]. Řešení zmíněných problémů se objevilo po roce 1968, kdy spatřilo světlo světa první PLC. Za otce tohoto vynálezu je považován Dick Morley [2].

1.1 Programovatelný logický automat

Programovatelný logický automat (PLC) je zařízení, které zpracovává a vyhodnocuje signály přicházející z technologických procesů, výrobních linek a strojů na jeho vstupy a v závislosti na vnitřním naprogramování zasílá povely na svoje výstupy. Těmito povely můžeme řídit například stykače motorů, ventily, klapky apod. Programovatelný logický automat je číslicově pracující elektronický systém vytvořený za účelem řízení průmyslových a technologických procesů, strojů a jiných logických úloh [4]. Největší význam mají PLC při řízení technologických celků, výrobních linek a strojů. Mezi výhody PLC patří jejich vysoká spolehlivost, snažší rozdělení řídicí struktury na samostatné celky s jasně definovatelnými rozhraními, nižší náklady na kabeláž apod. PLC nevyžadují příliš údržby, jejich program je snadno přestavitelný a díky tomu jej dokáže nastavit poměrně široké spektrum pracovníků [5]. PLC je složeno z centrální procesorové jednotky, systémové paměti, uživatelské paměti, vstupů a výstupů a komunikačních portů [4].

Struktura PLC

PLC je tvořeno:

- **Procesorem CPU**

Přijímá informace ze vstupního rozhraní, zpracuje je podle uloženého programu a aktualizuje je na výstupu. Zde probíhají úlohy firmwaru i samotného uživatelského programu [6].

- **Paměť**

Jde o operační paměť RAM a paměť programu ROM [6]. PLC od společnosti ABB používají paměť zálohovanou lithiovou baterií [7]. V případě odpojení napájení a baterie je potřeba aktualizovat hodnoty konstant.

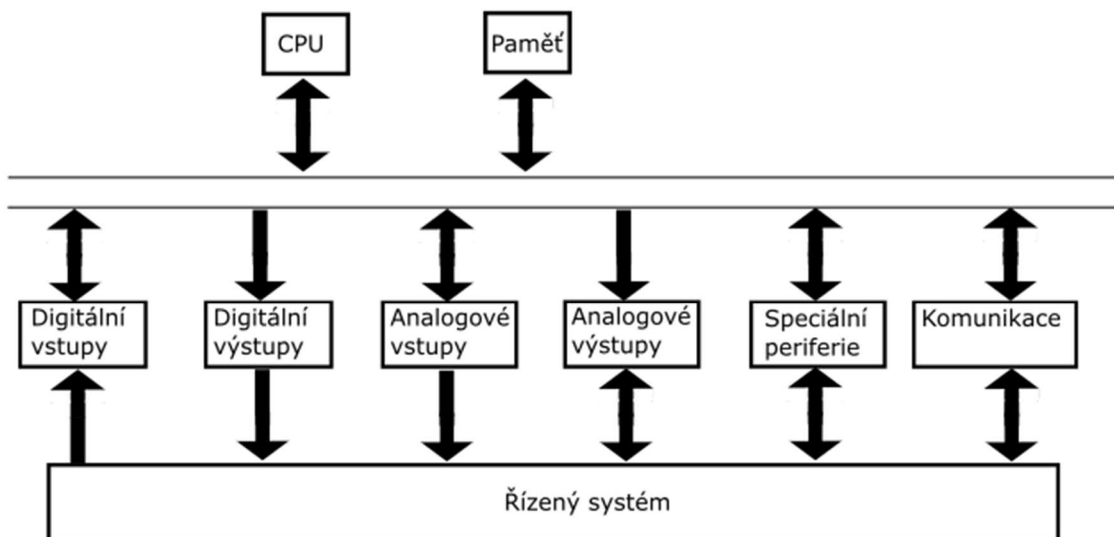
- **Systémová sběrnice**

Pomocí systémové sběrnice připojujeme vstupní a výstupní moduly [8]

- **Vstupní a výstupní moduly**

Tyto moduly mohou zpracovávat diskrétní nebo analogový signál. Tento signál vstupuje například ze snímačů, tlačítek nebo přepínačů a je převeden do formy potřebné pro aktivity CPU [6].

Blokové schéma PLC je znázorněno na Obr. 1.



Obr. 1: Struktura PLC (překresleno z [9])

2 Systémy průmyslové automatizace

Před zavedením samotného systému do provozu, je nutné vybrat vhodný typ. Dle možnosti změn v nastavení systému rozeznáváme tři základní typy systémů automatického řízení:

Pevná automatizace

Program je v takovém případě nastaven hardwarem a nelze jej snadno měnit. Výhodou takové automatizace je její velká rychlost, nevýhodou pak pořizovací náklady a obtížnost změny programu. Příkladem pevné automatizace jsou posuvné pásy ve výrobních halách nebo výroba jednoho typu výrobku ve velkých sériích. Pevná automatizace má svůj program definován vnitřní strukturou hardwaru [10]. V případě použití FPGA (Field Programmable Gate Array) založené na technologii antipojistek není možné již program překonfigurovat. Pro možnost přeprogramování lze využít FPGA založené na technologii SRAM (Static Random Access Memory) [11].

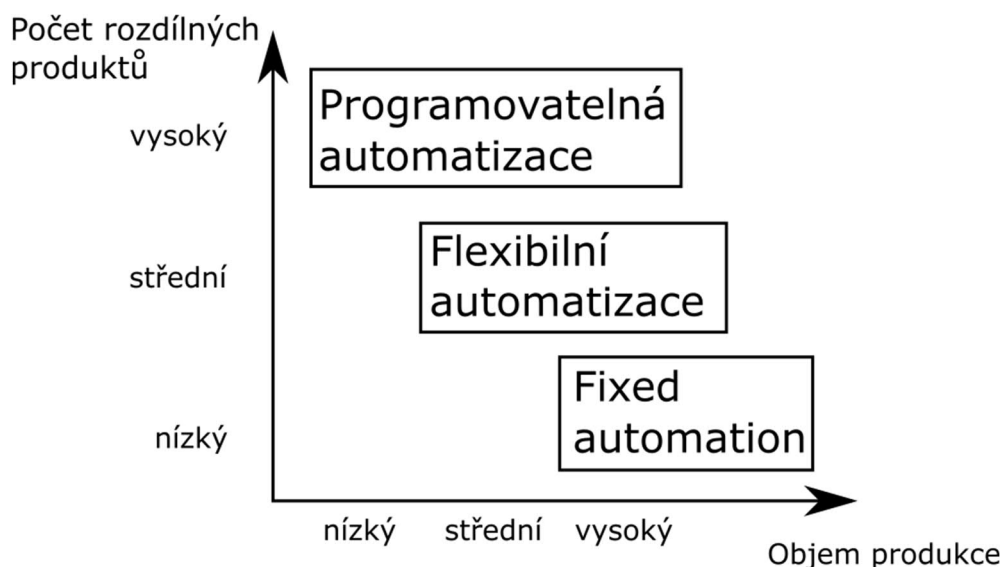
Programovatelná automatizace

Tato forma je vhodná pro výrobu produktů v dávkách od několika desítek kusů až po několik tisíc kusů. Pro každou novou dávku je navržen nový program. Rychlost programovatelné automatizace bývá obecně nižší než u pevné automatizace. Nevýhodou programovatelné automatizace je odstávka systému mezi změnou jednoho programu na druhý. Příkladem je číselně řízený obráběcí stroj nebo průmyslový robot [10].

Flexibilní automatizace

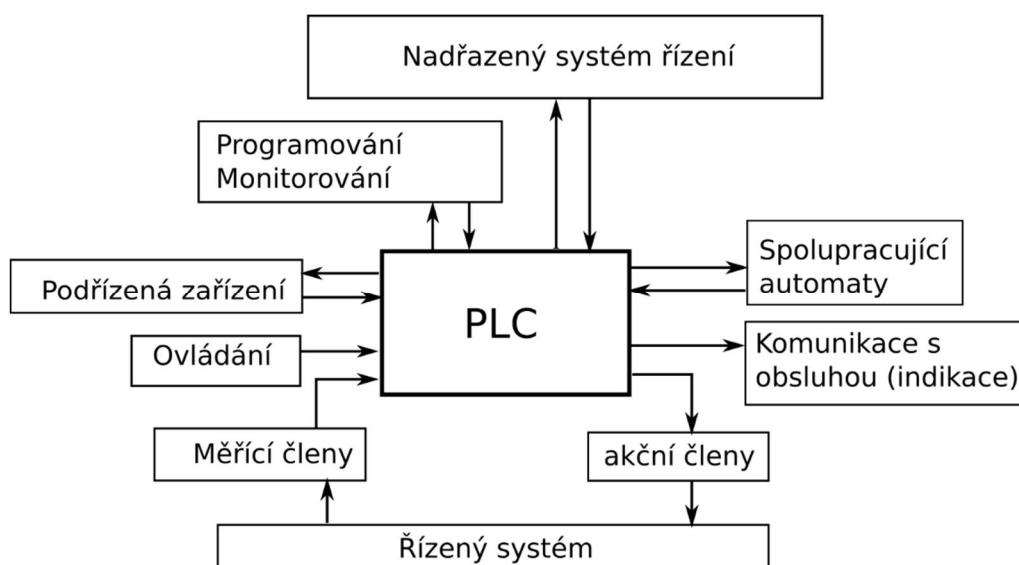
Má stejné schopnosti jako programovatelná automatizace, ale neztrácí čas při změně programu. Může tedy pracovat s celou řadou programů. Příkladem flexibilní automatizace jsou snadno roboti s možností pohybu ve více osách a s možností rychlého přeprogramování programu [12].

Při výběru vhodného systému se tedy zajímáme o to, jak často bude potřeba jeho program přestavovat a pro kolik produktů bude toto nastavení platné. Princip výběru je znázorněn na Obr. 2[12].



Obr. 2: Výběr vhodného systému (překresleno z [12])

Systém je soustavou složenou z více prvků. Jednotlivé prvky systému jsou vzájemně ovlivňovány a dochází mezi nimi k výměně informací energie nebo hmoty [13]. Mozkem systému automatického řízení může být právě PLC. Blokové schéma řízení s PLC je znázorněno na Obr. 3.



Obr. 3: Blokový schéma řízení s PLC (překresleno z [9])

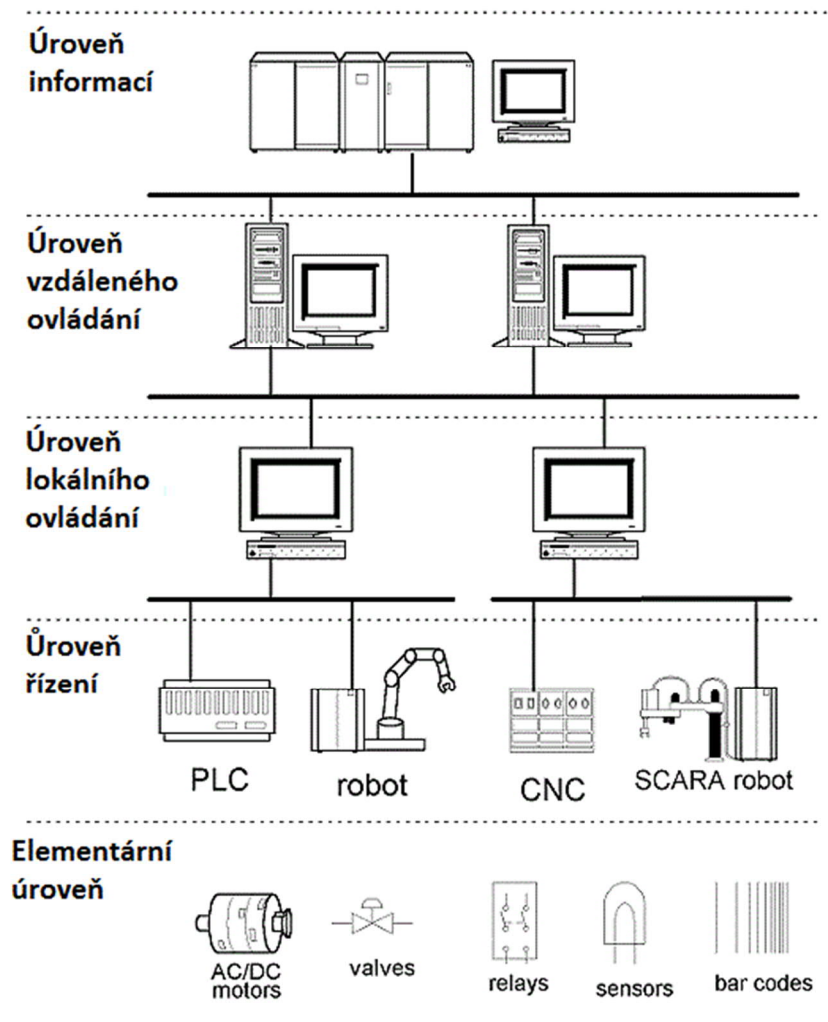
Rozsáhlejší systém automatického řízení je systém, který bývá složen z několika následujících prvků [9]:

- Snímače,
- prostředky pro přenos a uchování informací,
- převodníky,

- počítačové sítě,
- elektrické zesilovače,
- pneumatické a hydraulické zesilovače,
- programovatelné automaty,
- akční členy (Pneumatické a hydraulické motory, elektrické motory, ovládané ventily, čerpadla... [9]

2.1 Hierarchie systému průmyslové automatizace

Průmyslové automatizační systémy mohou být velmi složité, s velkým počtem zařízení pracujících s automatizačními technologiemi [14]. Na Obr. 4 je znázorněno hierarchické uspořádání automatizačního systému sestávajícího z různých úrovní.



Obr. 4: Hierarchické uspořádání systému průmyslové automatizace (Přeloženo z [14])

Elementární úroveň

Nejnižší úroveň hierarchie automatizace zahrnující zařízení jako jsou snímače nebo akční členy. Hlavním úkolem těchto zařízení je přenášet data procesů a strojů. Sensory přeměňují parametry jako je teplota, průtok, hladina, atd. v reálném čase na elektrické signály [14].

Úroveň řízení

Zde jsou například roboti, PLC automaty nebo CNC stroje. Jejich úkolem je získávat parametry z procesu z různých senzorů a pohánět pohony a jiné na základě zpracovaných signálů [14].

Lokální a vzdálené ovládání

V této úrovni se nacházejí monitorovací systémy, systémy ovládání a nastavení funkcí. Jedná se o rozhraní HMI (Human Machine Interface), systémy řízení a distribuce (DCS). Stanice mohou být místní nebo vzdálené [15].

Úroveň informací

Nejvyšší úroveň průmyslové automatizace, často nazývána také úroveň informací. Tato úroveň řídí celý systém automatizace. Na této úrovni je zpracován plán výroby, analýza trhu nebo objednávky a prodej [14, 15].

2.2 Centralizované a distribuované systémy

Při zavádění systému automatizace je nutné se rozhodnout, zda je pro vybraný celek vhodnější centralizovaný nebo distribuovaný systém řízení.

Centralizovaný systém

Je takovým systémem, kde jsou jednotlivé aspekty procesu řízeny z jediného centrálního CPU [16]. Výpočetní kapacita proto musí být výrazně vyšší. Výhodou je fakt, že existuje pouze jeden CPU - což znamená, že je zapotřebí pouze jedna taková náhradní část [17]. Výhodou je také jednoduchost hierarchie systému. Nevýhodou centralizovaného řízení může být ale pomalá rychlost komunikace s centrálním počítačem. Nevýhodou je také složitější větvení programu, protože na to je připraven lépe distribuovaný systém [18].

Distribuovaný systém

V distribuovaných řídicích architekturách je proces řízen větším počtem řídicích modulů. Každý modul má řídicí jednotku a systém obvykle obsahuje centrální řídicí PLC, který se

zabývá úkoly, jako je logistika, sledování součástí a statistika. Moduly PLC automatizují příslušné oblasti technologického celku [17]. Výhodou distribuovaného systému je možnost rozdělení složitého celku na menší dílčí celky. Je zde tak možnost programovat jednodušší programy odděleně a tyto programy dohromady řídí celý technologický celek. U distribuovaného systému také nejsou tak vysoké nároky na výpočetní výkon CPU, protože výpočet probíhá ve více procesorových jednotkách [19].

Skutečné systémy obecně nejsou úplně centralizovány nebo distribuovány. Ve většině případů jde o kombinaci obou typů. Při výběru vhodného systému je za potřebí zkoumání procesního zařízení, které řídíme. Musíme zvážit jak provozní aspekty zařízení, tak fyzické umístění kontrolovaných objektů a snímačů [16].

2.3 Komunikace v průmyslové automatizaci

Komunikace v průmyslové automatizaci často využívá průmyslových sběrnic. Tyto sběrnice jsou digitální sériové sběrnice zajišťující síťovou komunikaci v průmyslových rozvodech. Průmyslové sběrnice byly vyvinuty v 80. letech 20. stol. [20]. Důvodem pro zavádění průmyslových sběrnic byla snaha zlevnit kabeláž mezi instrumentací a řídicími členy, zmenšit počet drátů a zjednodušit projekt kabeláže [5]. V průmyslové automatizaci je využíváno několik desítek různých standardů (PROFIBUS, PROFINET, CANbus, Modbus, Foundation Fieldbus, DeviceNETAS-i atd.) [21, 22]. V mé práci bude využito komunikace za pomoci průmyslové sběrnice Profibus.

2.3.1 Profibus

Profibus je složitý protokol vytvořený na třívrstevném referenčním modelu komunikace využívající OSI (Open System Interconnection) referenční model [23]. Profibus (Process Fieldbus) byl vyvinut s podporou několika významných německých firem pod koordinací DBFT (Deutsches Bundesministerium für Forschungs und Technik). Stal se německou normou DIN 19245 [5]. Profibus má více variant komunikačních protokolů - FMS (Fieldbus Message Specification), PA (Process Automation) and DP (Decentralized Periphery) [23]. V této práci bude využit PROFIBUS DP.

Profibus DP

Profibus DP (Decentralized Peripherals) je hojně využíván v zařízeních průmyslové automatizace, výrobních linkách a CNC strojích. Do této sběrnice jsou připojeny periferie IO, řídicí automaty, frekvenční měniče, operátorské panely apod. V jediném segmentu může být

připojeno až 32 stanic. Za segment je považován úsek kabelu PROFIBUS, který je zakončen tzv. zapnutými odpory. Pro přechod do dalšího segmentu je použit repeater – opakovač signálu. Kabel je složen z dvou vodičů (červený a zelený), které umožňují přenos informací. Mezi vodiči a vrchním pláštěm je stínění. Metalické vedení je připojeno prostřednictvím rozdílové napěťové sběrnice podle standardu RS 485. Přenos informace může dosahovat rychlosti až 12Mb /s. Různé rychlosti sítě PROFIBUS jsou omezeny maximální délkou jednoho segmentu [23]. Maximální délka přenosu je 1,2 km pro metalický a 80 km pro optický přenos, pak je nutno zařadit repeater [24]. Protokol rozlišuje následující typy zařízení: Master a Slave [23, 25].

Řídící Master stanice

Sít' Profibus DP je řízena stanicí „MASTER“ (tzv. MONO-MASTER nebo také aktivní stanice). Master řídí tok dat komunikace sběrnice. Jedna řídicí síť může obsahovat více „MASTER“ stanic (tzv. Multi-Master). Po ukončení komunikace jednoho „MASTER“ členu, je uvolněno řízení sběrnice pro další řídicí členy. V tomto projektu je master stanicí PLC automat [23, 25].

Stanice Slave

Slave zařízení jsou periferie jako například digitální vstupy, digitální výstupy, ventily, pohony a měřicí senzory. Slave může pouze potvrdit přijetí zprávy. Slave stanice jsou například aktuátory a snímače [23].

Zakončení

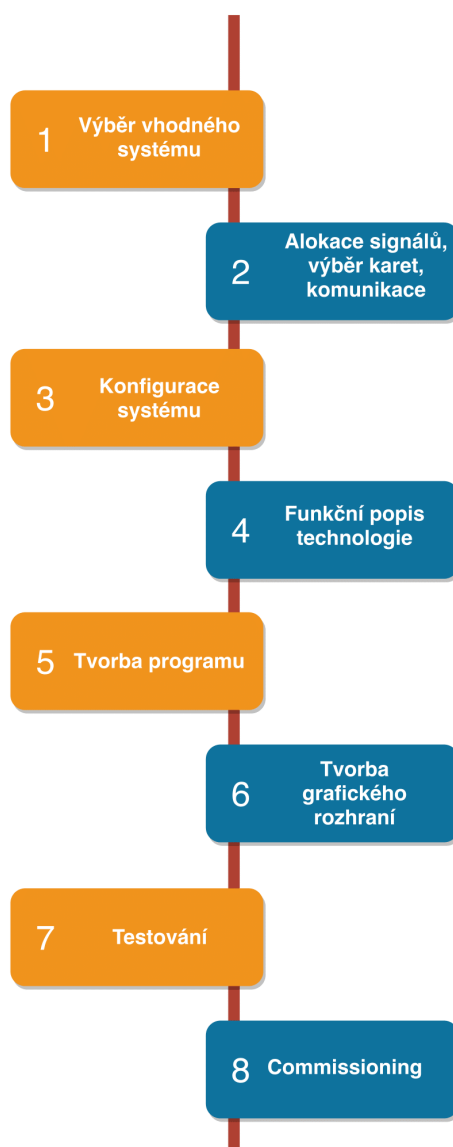
Na konci sběrnice je umístěn aktivní „bus terminátor“, jehož úkolem je udržet úroveň signálu na předem stanovené úrovni. Dále pak zajišťuje přizpůsobení impedance a zabránění nežádoucích odrazů signálu [25].

3 Návrh systému řízení reálné úpravy vod

Součástí diplomové práce je návrh reálného řídicího systému pro část technologického celku úpravy vod. Tento návrh je popsán v následujících kapitolách.

3.1 Proces návrhu řídicího systému

Jednotlivé dílčí kroky návrhu jsou v této kapitole zaznamenány do procesu, který jasně definuje posloupnost úkolů. Tento proces je platný v situaci, kdy je již samotný celek fyzicky sestaven. Proces je slovně popsán níže a jeho grafická reprezentace je zobrazena na Obr. 5.



Obr. 5: Proces tvorby systému

Prvním krokem je výběr vhodného systému pro řízení celku dle rozsahu celku a požadavků na spolehlivost systému, počet operátorských pracovišť, počet signálů, popřípadě požadavků rychlost systému a další. Druhým krokem procesu je alokace I/O signálů na karty a také výběr komunikačních protokolů. Třetím krokem je konfigurace systému. V této fázi jsou do systému přidány použité kontroléry, vybrané I/O karty a komunikace. Čtvrtým krokem je pochopení technologie. V tomto kroku může být vhodné připravit si vývojové diagramy. U větších celků to bohužel není příliš možné (vývojových diagramů by bylo příliš) a vytváří se tak jen funkční popisy. V případě předělávání starého již funkčního celku do nového systému by mohl programátor použít pro pochopení funkce logiky ze starého systému. V pátém kroku je vytvořen program za pomoci vybraných metod programování. V následujícím šestém kroku jsou kresleny displeje pro operátory. Posledním v této práci provedeným krokem je šestý krok – testování. V této fázi ještě nejde o testování na skutečném celku. Pokud je tento proces hotov je možné začít proces testování na skutečném celku a následné uvádění technologie do provozu. Tento poslední krok je nazýván jako commissioning. Vzhledem k faktu, že tento krok probíhá na skutečné stavbě, není součástí méj diplomové práce.

3.2 Popis technologického celku

Navrhovaný systém představuje řešení pro řízení technologického celku, který je součástí čistírny vod v Norském městě Fredrikstad. Piping and instrumentation diagram čistírny je zobrazen v příloze (Příloha 1) a popsán v kapitole 3.3. Tato kapitola je věnována obecnému popisu celé čistírny, zatímco kapitola 3.6 (Technologický popis zpracovávaného celku) popisuje pouze část, které je tato práce věnována. V současnosti se čistírna využívá na hranici svojí kapacity – očistit vodu pro 24 000 osob a z tohoto důvodu je rozšiřována. Čistírna vod by měla být schopna po dokončení očistit vodu pro 48 000 obyvatel. Čistírna pracuje na mechanicko-chemickém principu čištění [26]. Čistírna je tvořena následujícími částmi:

3.2.1 Oddělovač před čistírnou

Objekt zabraňuje, aby se do čistírny dostalo takové množství odpadních vod, které překračuje její kapacitu. Oddělovač tvoří část přívodní stoky upravené tak, že v úrovni odpovídající maximálnímu průtoku čistírnou je osazena stavitelná přepadová hrana, přes kterou přebytečná voda přepadne a bez čištění odeče [27].

3.2.2 Lapák štěrku

Objekt slouží k zachycení hrubých předmětů na dně přívodní stoky. Tvoří jej železobetonová jímka situovaná na přivaděči těsně před čistírnou. Doporučuje se vyklízení drapákem na přilehlou zpevněnou plochu [28].

3.2.3 Česle

Zařízení sloužící zachytávání hrubých nečistot a plovoucích objektů. Tvoří je mřížoví ocelových prutů (česlic), které je vloženo do celého průtočného profilu přítokového žlabu za lapák štěrku. Česle jsou strojně stírané. Odstraňování shrabků se děje pásovým nebo šnekovým dopravníkem do přistaveného kontejneru [28].

3.2.4 Lapák písku

Lapáky písku jsou určeny pro předčištění odpadních vod z městských a průmyslových kanalizací a slouží pro zachycení písku. Do lapáku vstupuje voda, již hrubě mechanicky předčištěna, zbavena štěrku a jiných větších objektů. Lapáky jsou předřazeny před čistírny odpadních vod, nebo zařízení, do kterých má natékat voda zbavená písku. Písek je třeba odstranit z proudu odpadní vody proto, aby nesedimentoval a nehromadil se v dalších částech čistírny. Principem odstraňování těchto látek je snížení průtočné rychlosti v nádrži, v důsledku kterého dochází k sedimentaci zrněk písku, avšak nedojde k usazení ostatních organických látek [29].

Písek je dopravován do pískových kapes. Písek je pohybován ve směru vpřed a vzad. Rychlost zpětného pohybu je asi 2,5 krát rychlejší než rychlost pohybu vpřed. Písková čerpadla čerpají písek z pískových jam do kanálů. Dokud je lapač v provozu, pracují nepřetržitě dmychadla na provzdušňování. Vzdušněním se vytváří rotace kapaliny a tato přispívá k lepšímu oddělení částic organického původu s nižší hmotností, které jsou vyplaveny vodou [30].

3.2.5 Lapáky tuků a olejů

Lapač tuků následuje za lapákem písku, odstraňuje tuk/kal v nádrži. Do nádrže je přiváděn pod tlakem vzduch. Na vzduchové bublinky se nalepují tukové částice a jsou vnášeny na povrch, kde jsou stírány do sběrného žlabu [27].

3.2.6 Usazovací nádrže

Slouží k separaci tuhé fáze od kapaliny. Slouží k zachycení dobře usaditelných nerozpuštěných látek před dalšími stupni čištění (součást mechanického stupně čištění). Používají se, aby odlehčily zatížení druhého (biologického) stupně čištění [28].

3.2.7 Biologické čištění

Začíná po mechanickém předčištění. Mikroorganismy rozkládají neusaditelné látky, redukují a stabilizují organické látky a odstraňují dusík a fosfor. Technologie napodobuje samočisticí proces vyskytující se v přírodě [28].

3.3 Piping and instrumentation diagram

V příloze (Příloha 1) je znázorněno schéma celé čistírny vod. Takové schéma bývá známé pod názvem piping and instrumentation diagram (P&ID). Část technologického celku, pro který je systém navrhován, je ohraničena červeným rámečkem.

3.3.1 Systém značení

V P&ID je mnoho objektů. Symboly těchto objektů a písmena uvnitř těchto symbolů vycházejí z normy (ISO 15519-2:2015). Identifikační kód může být složen až z pěti písmen. První písmeno definuje měřenou veličinu. Druhým písmenem je tzv. modifikátor, tím může být například poměr (F – fraction) nebo časová rychlost změny (K – Time rate of change). Třetí písmeno indikuje čtecí/ pasivní funkci. Čtvrté písmeno představuje výstupní/ aktivní funkci. Posledním písmenem je Funkční modifikátor, který reprezentuje signalizovaný stav (například High, closed, open). Například LAL – Level alarm low [31, 32]. Názvy objektů jsou umístěny nad objektem a v tomto projektu jsou složeny ze dvou částí. První částí jsou tři písmena reprezentující zkrácený název technologie, ve které je objekt umístěn a jeho číselné označení. Například SAN1-XXX odkazuje na sandfang, tedy lapák písku. Druhou část (za pomlčkou) tvoří dvojice písmen a číselné označení. V této části jde o typ zařízení (PU – pump). Například SAN1-PU01 je pumpa, která je součástí technologie lapáku písku. Značení signálu je pak stejné jako značení objektů, je zde ale jedna část navíc. V této další části (odděleno opět pomlčkou) je odkázáno na povahu signálu. Identifikační kód definuje měřenou/signalizovanou veličinu. Celý signál pak může vypadat takto SAN1-PU01-SS1 (Sandfang-pumpe-sikkerhetsbryter = v překladu Lapák písku – pumpa – bezpečnostní spínač) [31, 32].

3.3.2 Výčet objektů

V tabulce (Tabulka 1) Tabulka 1 je seznam objektů, které jsou součástí technologického celku, na který je tato diplomová práce zaměřena. Původní názvy objektů jsou v norštině. V norštině bylo napsáno zadání projektu a tyto názvy jsou v tabulce ponechány, protože jsou shodné s názvy v příložené zadávací dokumentaci.

Tabulka 1: Seznam objektů [zdroj: Funkční popis projektu - přidán do přílohy]

objekt	Původní název	český překlad
LTL1-AV01	Kjølevifte blåsemaskin sandfang 1	Chladicí ventilátor lapače písku 1
LTL1-AV02	Kjølevifte blåsemaskin sandfang 2	Chladicí ventilátor lapače písku 2
LTL1-AV03	Kjølevifte blåsemaskin sandfang standby	Chladicí ventilátor lapače písku záložní
LTL1-BM01	Blåsemaskin til sandfang 1 Frekvensstyrt	Dmychadlo lapáku písku 1 frekvenčně řízený
LTL1-BM02	Blåsemaskin til sandfang 2 Frekvensstyrt	Dmychadlo lapáku písku 2 frekvenčně řízený
LTL1-BM03	Blåsemaskin til sandfang standby Frekvensstyrt	Dmychadlo lapáku písku záložní frekvenčně řízený
LTL1-FT01	Mengdemåler sandfang 1	Měřič průtoku 1 v lapáku písku
LTL1-FT02	Mengdemåler sandfang 2	Měřič průtoku 2 v lapáku písku
SAN1-ML01	Motorluke utløp sandfang 1 Matic	Motorový uzávěr (poklop) lapáku písku 1
SAN1-PU01	Sandpumpe 1 Frekvensstyrt	Písková pumpa 1 frekvenčně řízená
SAN1-PV01	Pn.ventil sand fra sandfang 1	Pneumatický ventil 1 v lapáku písku
SAN1-SK01	Sandskrape 1 Frekvensstyrt	Škrabák písku 1 frekvenčně řízený
SAN2-ML01	Motorluke utløp sandfang 2 Matic	Motorový uzávěr (poklop) lapáku písku 2
SAN2-PU01	Sandpumpe 2 Frekvensstyrt	Písková pumpa 2 frekvenčně řízený
SAN2-PV01	Pn.ventil sand fra sandfang 2	Pneumatický ventil 2 v lapáku písku
SAN2-SK01	Sandskrape 2 Frekvensstyrt	Škrabák písku 2 frekvenčně řízený
FET0-FT01	Mengdemåler fett til fra sandfang til innløp	Měřič množství tuku 1
FET0-FT02	Mengdemåler fett til blandeasseng	Měřič množství tuku 2
FET0-PV01	Pn.v. fett pumping til blandeasseng	Pneumatický ventil čerpání tuku do mísícího bazénu
FET0-PV02	Pn.v. vann pumping til før sandfang	Pneumatický ventil čerpání vody před lapačem písku
FET1-LT01	Nivå giver fettfang 1	Měření hladiny v zásobníku tuku 1
FET1-LT03	Nivå giver fettkum 3	Měření hladiny v zásobníku tuku 3
FET1-PT02	Trykk giver etter fettpumpe 1	Snímač tlaku tukové pumpy 1
FET1-PU01	Fettpumpe til blandeasseng/innløp Frekvensstyrt	Čerpadlo tuků na vstupu mísícího bazénu, frekvenčně řízený
FET1-PV02	Pn.v. fett pumping fra fettkum 1	Pneumatický ventil čerpání tuku z nádrže na tuky 1
FET1-PV03	Pn.v. slam fra blandeasseng til fettkum 1	Pneumatický ventil mezi mísícím bazénem a bazénem na tuky 1
FET1-RX01	Omrører i fettkum 1 Frekvensstyrt	Míchání tuku 1 frekvenčně řízený
FET1-SK01	Overflateskrape fettfang 1 Frekvensstyrt	Povrchový škrabák tuku 1 frekvenčně řízený
FET1-SV01	Magnetventil spylevann fettfang 1	Magnetický ventil vody na proplachování tukové pasti 1
FET1-SV02	Magnetventil trykkluftsomrøring fetttank 1	Magnetický ventil stlačeného vzduchu tukové nádrže 1
FET1_QT01	TS-måler fett fra fettkum 1	Korekční měřidlo změny hustoty 1
FET2-LT01	Nivå giver fettfang 2	Spínač hladiny lapače tuků 2

FET2-LT03	Nivågiver fettkum 2	Spínač hladiny 2
FET2-PT02	Trykkgiver etter fettpumpe 2	Snímač tlaku tukové pumpy 2
FET2-PU01	Fettpumpe til blandebasseng/innløp	Tuková pumpa na vstupu směšovacího bazénu
FET2-PV02	Pn.v. fettumping fra fettkum 2	Pneumatický ventil čerpání tuku z nádrže na tuky 2
FET2-PV03	Pn.v. slam fra blandebasseng til fettkum 2	Pneumatický ventil mezi mísícím bazénem a bazénem na tuky 2
FET2-RX01	Omrører i fettkum 2 Frekvensstyrt	Míchání tuků 2 frekvenčně řízeno
FET2-SK01	Overflateskrape fettfang 2 Frekvensstyrt	Povrchový škrabák tuku 2 frekvenčně řízený
FET2-SV01	Magnetventil spylevann fettfang 2	Magnetický ventil vody na proplachování tukové pasty 2
FET2-SV02	Magnetventil trykkluftsomrøring fetttank 2	Magnetický ventil stlačeného vzduchu tukové nádrže 2
FET1-OT02	TS-måler fett fra fettkum 2	Korekční měřidlo změny hustoty 2

Ve následující tabulce (Tabulka 2) je výčet opakujících se typů signálů.

Tabulka 2: typy signálů [zdroj: Funkční popis projektu - přidán do přílohy]

signál	Původní název	Překlad
XXX-XXX-AU1	Fjern-modus på fr.omf.	místní / vzdálené ovládání
XXX-XXX-DR1	drift	zpětná vazba Ready to operate
XXX-XXX-SS1	sikkerhetsbryter	spínač nouzového odstavení
XXX-XXX-UC1	pådrag fr.omf.	Setpoint otáček pro měnič
XXX-XXX-US1	start	povel pro zapnutí měniče
XXX-XXX-XA1	sikring utløst	pojistka přetavena
XXX-XXX-XA2	feil fr.omf.	chyba měniče
XXX-XXX-GS1	endebryter åpen	koncový spínač/otevřený stav
XXX-XXX-GS2	endebryter lukket	koncový spínač/uzavřený stav
XXX-XXX-PQ1	mengdepuls	měření pulsů
XXX-XXX-PV1	målesignal mengde	hodnota měřeného signálu

3.4 Základní požadavky na systém kontroly a řízení

Signály systému mají mít svoji redundantní zálohu. Systém jako celek musí být schopen pracovat s více jak 1500 signály (s redundancí přes 3000). Systém má být ovládán z více operátorských pracovišť a má mít také nadřazené pracoviště se vzdáleným přístupem. Z tohoto pracoviště je možno upravovat systém kontroly. Systém je ovládán z jednoduchých operátorských displejů. Na displejích je vidět v jakém stavu se nacházejí jednotlivá zařízení.

3.5 Hierarchie řídicího systému

Součástí projektové dokumentace bývá také schéma, ze kterého je patrné rozložení jednotlivých komponentů a zařízení, z pohledu hierarchie. V příloze (Příloha 2: Hierarchické uspořádání projektu) je vyobrazeno uspořádání pro realizovaný projekt. Na nejnižší úrovni jsou

I/O karty typu S800 komunikující přímo se zařízeními v poli [33]. Na nejnižší úrovni jsou ovládány také frekvenční měniče. Řešení zahrnuje ABB také bezpečnostní moduly ABB pluto [34]. Jednotky pluto si vyměňují data bez nutnosti programování dodatečné komunikace (až 32 jednotek). Na kontrolní úrovni je PLC automat PM866 [35]. Právě do tohoto automatu je nahráván program pro řízení. Nad tímto automatem je systémová síť, přes kterou probíhá například komunikace mezi operátorským pracovištěm a PLC automatem. Tato síť je zálohována a komunikace je tedy redundantní.

3.6 Technologický popis zpracovávaného celku

Diplomová práce je věnována části technologického celku (nacházející se v červeném rámečku v příloze 1), jejíž součástí je technologie pro odbourávání písku a tuků. Čistírna jako celek, má již před touto částí prvky pro hrubé předčištění vody a za touto částí další jiné výše popsané technologie.

3.6.1 Lapák písku

Lapák písku má být uveden do provozu, pokud proteklo 200 000 l na měření UTL0-FT01 nebo po uplynutí jedné hodiny od jeho posledního spuštění. Spuštěna je škrabka písku SK01 ve spodní části nádrže. Škrabka je tlačena po dně vpřed a poté se vrací zpět. Rychlost pohybu vpřed má být nastavena na 40 % a pro pohyb zpět 80 %. Vždy když je dosaženo výchozí polohy, je aktivován koncový spínač (SAN1_SK01_GS1) a je započten 1 cyklus. Pokud není od startu pohonu škrabáku do času T2 (přednastavená hodnota T2=2 minuty) dosaženo výchozí polohy, je indikován alarm. Počet těchto cyklů je potřeba nastavit na 5. Časový interval pro změnu rychlosti má být nastaven na určitou dobu od sepnutí koncového spínače domácí polohy. Výchozí hodnotou je 10 s ale přesný čas bude doladěn na místě. Škrabák je vybaven ochranou před přetížením (WT1). Ochrana WT kontroluje hodnotu proudu analogového signálu, na jeho základě je možno signalizovat dva alarmy. Při pohybu škrabáku vpřed je první alarm ochrany nastaven na 13,92 mA a druhý na 14,4 mA. Při návratu škrabky je první alarm signalizován při 9,6 mA a druhý 10,08 mA. Pokud hladina v nádrži klesne na 0,3 m (signál FET1-LT01), má být spuštěn pohon ML01, který zapříčiní otevření poklopu pro průtok vody. Je potřeba, aby pohon pro otevření poklopu byl aktivní po dobu 1 min. Pokud je spuštěn lapák písku, jsou spuštěny ventilátory na provzdušňování. Ventilátory na provzdušňování BM01 až BM03 je potřeba nastavit na 80 % výkonu. Počet cyklů škrabáku před čerpáním pumpy PU01 je potřeba nastavit na 10. Čerpání začíná tím, že se po dosažení cyklů škrabáku otevře ventil PV1. Dříve než přijde signál, že je ventil otevřen, nesmí být čerpání spuštěno. Pumpa má být zapnuta 20

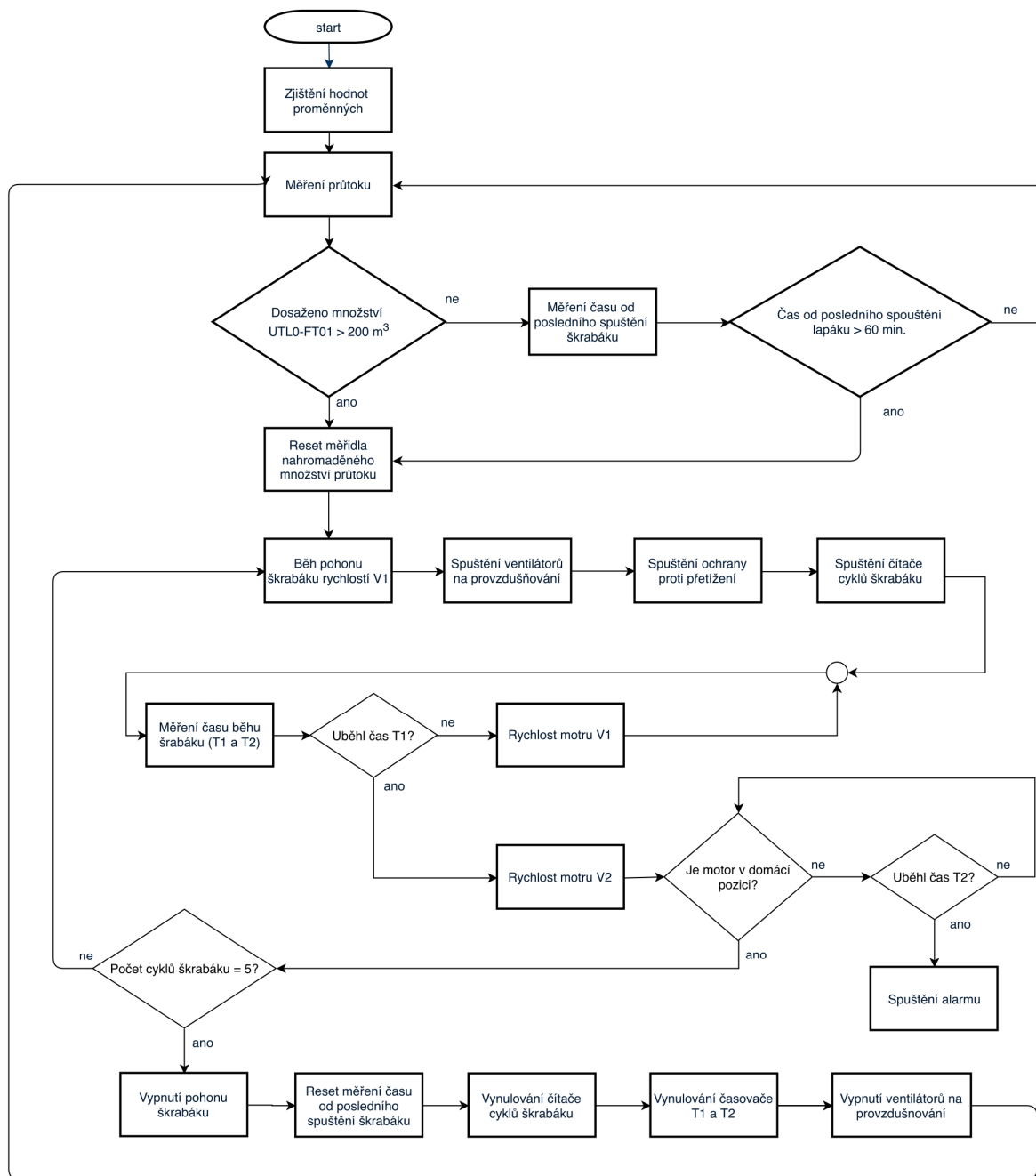
sec. na 80 % výkonu. Pumpy SAN1_PU01 a SAN2_PU02 neběží současně. V případě, je požadavek na sepnutí obou současně, čeká druhá, než skončí práce první. Ventil PV01 uzavírá nebo otevírá cestu písku z nádrže na písek. Má být nastaven tak, aby se zavřel 10 s po zastavení pumpy PU01 pro odtok písku.

3.6.2 Škrabák tuku

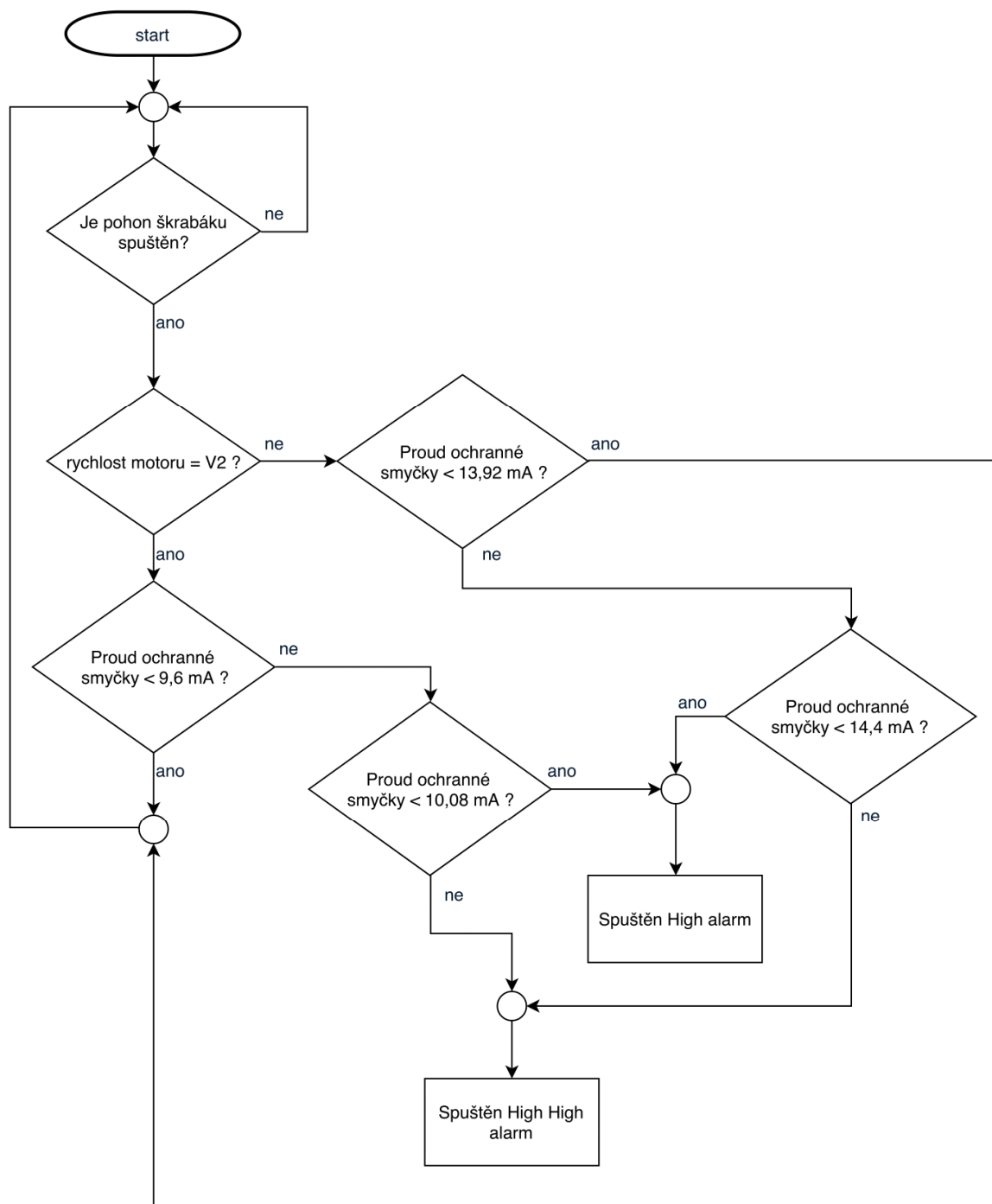
Škrabák tuku má být uveden do provozu, pokud proteklo 200 000 l na měření UTL0-FT01 nebo po uplynutí jedné hodiny od jeho posledního spuštění. Škrabák tuku má být uveden do provozu, pokud je zaznamenán na průtokoměru UTL0-FT01 průtok 200 l/s. Pokud není lapák aktivní po dobu jedné hodiny, je běh škrabáku vynucen. Pohon povrchové stěrky škrabáku má být nastaven na 75 % otáček. Škrabák se pohybuje směrem vpřed a na jeho konci aktivuje koncový spínač a vrací se do výchozí pozice, ve které je proplachován. Počet cyklů škrabáku je nastaven na 1. Počet provozních cyklů před propláchnutím povrchových škrabáku má být nastaven na 3. Doba proplachování povrchového škrabáku má být 5s. Jeden cyklus je započítán pokaždé, když je spuštěn koncový spínač GS1. Proplachování je spuštěno otevřením ventilu SV01. Ventil SV01 smí být otevřen jen pokud neběží škrabák FET1-SK01. Pokud dosáhne hladina vody v tukové nádrži (LT03) 2,5 m, je potřeba spustit odčerpávání tukové vody. Čerpadlo PU01 má být spuštěno na 75 % otáček. Čerpání se zastaví, když korekční měřič FET1-QT01 detekuje změnu hustoty čerpaných médií (hustota větší než 30 %) nebo bylo dosaženo bezpečnostního limitu hladiny pro zastavení čerpadla (1 m na FET1_LT03). Po zastavení čerpadla FET1-PU01 musí být zavřeny ventily FET1-PV02 a FET0-PV02. Po zaznamenání poklesu hladiny na 1 metr má být také spuštěno míchání kalové směsi.

3.7 Funkční popis

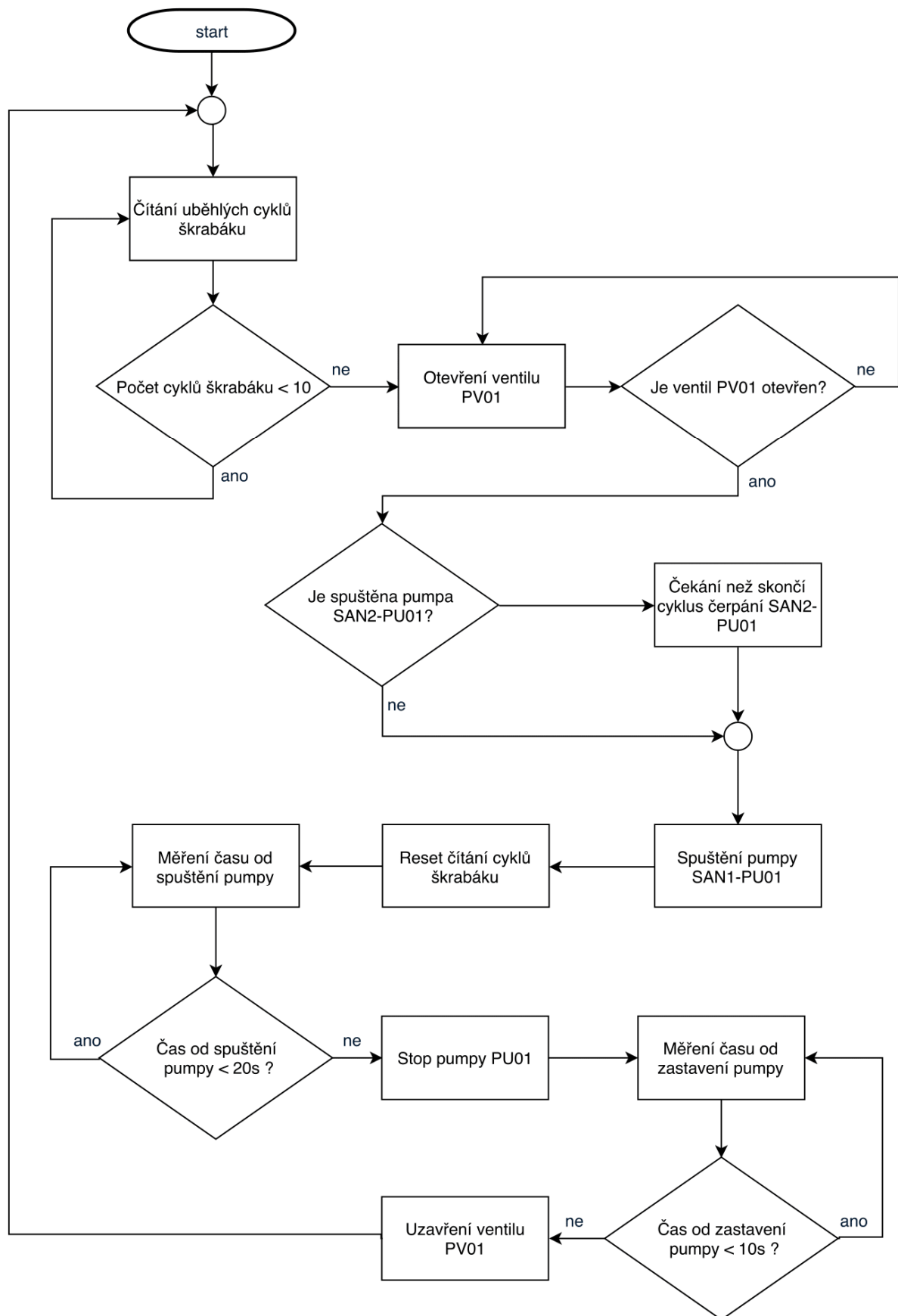
Pro přehlednější pochopení funkce technologie jsou v některých případech vytvářeny vývojové diagramy. Vytvoření takového diagramu usnadní programátorovi práci při následném programování. Vývojový diagram je grafickým znázorněním jednotlivých kroků algoritmu. Pro technologii zachytávání písku byly vývojové diagramy vytvořeny. Následující obrázky představují vývojové diagramy jednotlivých částí technologie.



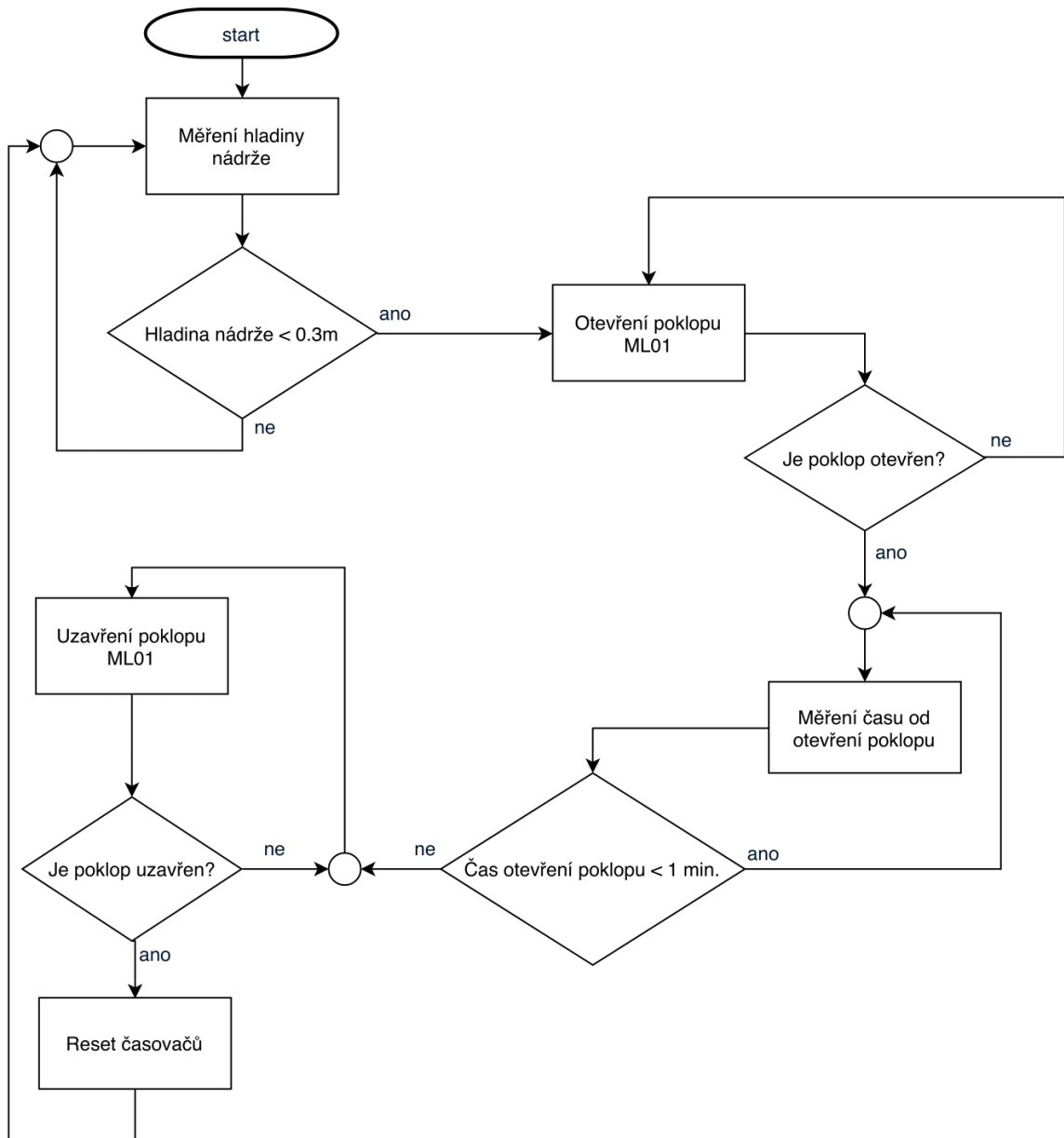
Obr. 6: Vývojový diagram pro lapák písku



Obr. 7: Vývojový diagram ochrany před přetížením škrabáku



Obr. 8: Vývojový diagram funkce čerpání pump



Obr. 9: Vývojový diagram funkce poklopu

3.8 Konfigurace systému

Konfigurací systému je myšlen výběr vhodného systému, alokace signálů na I/O karty, výběr karet, konfigurace softwaru, kde je nutno nastavit rozsahy signálů a přiřadit tyto signály na karty podle plánované alokace a dále také způsob komunikace. Před zahájením konfigurace je tedy potřeba mít připravený list se signály, které budou v programu použity. Pokud je list se signály k dispozici, je možno začít s jejich alokací.

3.8.1 Alokace signálů

Nyní je na řadě zjistit kolik vstupních nebo výstupních karet budeme potřebovat. O tomto rozhoduje nejen počet HW signálů, ale také je potřeba dodržet jisté zásady redundance. V průmyslové automatizaci, totiž ve většině případů jeden signál zastupuje druhý, stejně tak i jedno zařízení má svoji zálohu ve druhém, které je připraveno nahradit práci prvního. Příklad je vidět na Obr.10 kde jsou primární dmychadla (vlevo) i jejich záloha (vpravo).



Obr. 10: Příklad zálohovaného zařízení

Pokud je primární signál sveden na stejnou kartu jako jeho záloha, tak je spolehlivost redundance ztracena, protože stačí selhání jedné karty a jsou přerušeny oba signály. V ideálním případě se redundantní signál alokuje na jiný cluster než primární signál, avšak ve skutečnosti mohou hrát roli i finance na projekt a signál je sveden na jinou kartu stejného clusteru. Alokaci signálu pak myslíme jeho přiřazení na konkrétní kanál v dané kartě v rozvaděči. Po dokončení této přípravné práce má každý HW signál svoje místo a na kartách v rozvaděči je ideálně 20 % volného místa jako tzv. kanálová rezerva. Pro komunikaci mezi frekvenčními měniči a PLC je použit profibus (popis profibusu je obsažen v kapitole 2.3.1 a implementace komunikačního protokolu do systému pak v kapitole 4.5.5). Po dokončení této práce vznikla tabulka I/O signálů s konkrétní adresou na konkrétní kartě. Tato tabulka je přílohou na samostatném nosiči.

3.9 Vybraný systém řízení

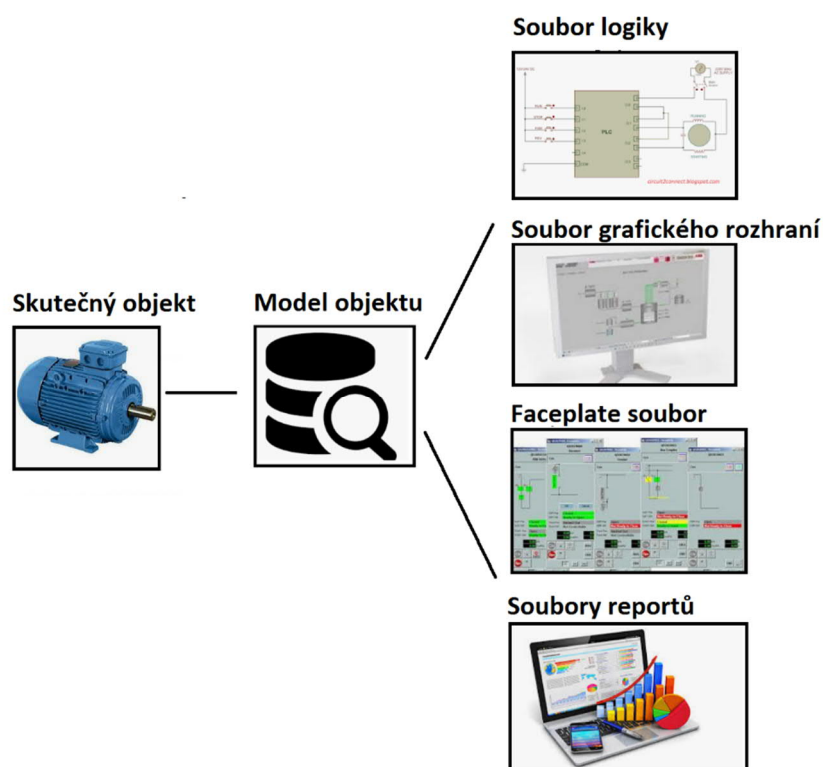
Vybraným systémem je systém 800xA od společnosti ABB. Čistička pracuje s 3000 signály a je ovládána z několika operátorských pracovišť. Vzhledem k požadavku na ovládání z více míst a potřebě pracovat s velkým (tisíce) množstvím signálů byl vybrán systém ABB 800xA. Tento systém je označován jako DCS (distributed control system) [36]. Za DCS systém je považován systém kompletního řešení automatizace s možností decentralizace jednotlivých částí řízení, splňující vysoké požadavky na redundanci, s možností integrace více aplikací a s rozhraním HMI vizualizace [37].

3.9.1 Systém 800xA

Jedná se o systém představující rozšířenou automatizaci. Rozšiřuje dosah tradičních systémů automatizace o možnost kontroly procesu, zvýšení energetické účinnosti a úspory energie. Systém i jeho síť je redundantní a vysoce spolehlivý. Byl vytvořen pro integraci systémů, aplikací a zařízení. V tomto systému jsou integrovány řídicí programy, operátorské HMI grafická rozhraní, alarmy, hardware a další. Řídicí programy – logiky tohoto projektu jsou vytvářeny v programu control builder, ale vidíme je i z prostředí 800xA, protože do tohoto inženýrského prostředí jsou integrovány [38]. Komunikace mezi kontroléry a systémem je zajištěna za pomoci OPC serveru. OPC server je součástí systému 800xA [39].

3.9.2 Koncept objektu a aspektu

Systém je postaven tak, že v **Control Builderu** je otevírán daný projekt například za účelem editace jeho programu, alarmů, testování programu a podobně, v **Graphics Builderu** (nebo v programu Visual basic) jsou vytvářeny HMI displeje, zatímco v inženýrském prostředí – **Engineering workplace** vidíme všechny projekty a jejich objekty nahrané v celém **systému 800xA**. Jeden objekt je tak reprezentován více soubory, tak jak je vidět na obrázku (Obrázek 11) [38]. Takové soubory nazýváme aspekty a jeden objekt tak obsahuje více aspektů [40].

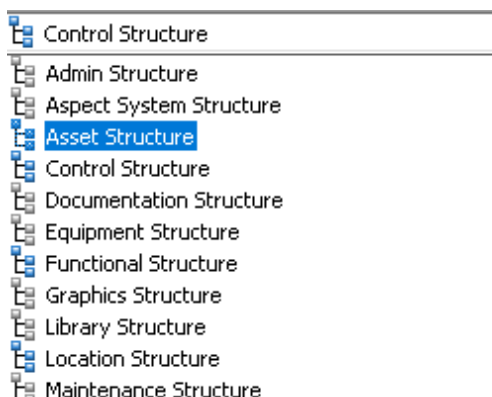


Obrázek 11: Koncept objektu a aspektu (Překresleno z [41])

Faceplate je grafickou reprezentací vlastností objektů s možností pro interakci operátora [42]. Aspekty jsou rozčleněny do jednotlivých struktur.

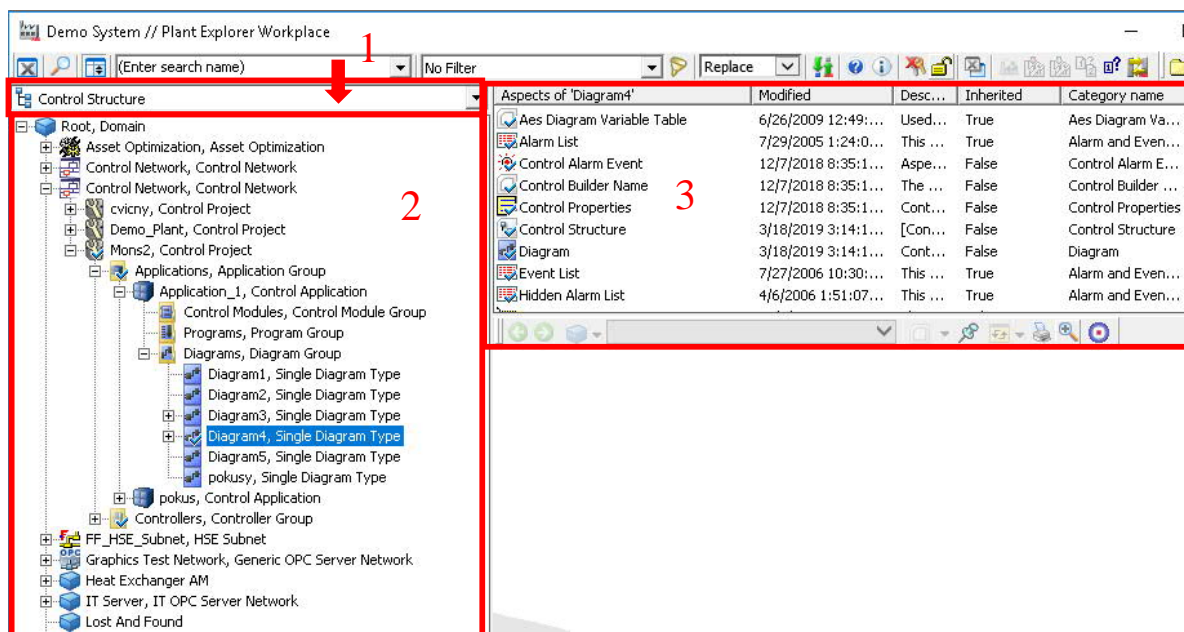
3.9.3 Struktury Systému

Vytvořené objekty a aspekty mají reprezentovat celý řízený systém. Přirozená cesta k reprezentaci povahy jednotlivých objektů a aspektů v systému je jejich rozdělení do struktur. Jeden objekt je možné vidět i z více struktur. Systém obsahuje v základu 19 struktur, některé z nich je možné vidět na Obr. 12 [43].



Obr. 12: Náhled do struktury systému 800xA

K procházení struktur byl vyvinut **Plant Explorer**, který je součástí Systému 800xA. Rozhraní Plant Exploreru je rozděleno do několika oken (Obr. 13). V okně 1 je vybírána uživatelem struktura do které chce nahlížet. V okně 2 jsou vidět objekty ve vybrané struktuře. V okně 3 jsou vidět aspekty vybraného objektu.



Obr. 13: Plant Explorer

Uživatelé systému mají různé role a podle toho také různá oprávnění. Například operátor může mít zajištěn přístup jen k HMI grafickému rozhraní. Systém umí pracovat s velkým množstvím komunikačních protokolů, a proto je do něj integrováno téměř každé zařízení z celku. Inženýrské prostředí zobrazuje informace v reálném čase. Možnosti systému jsou rozsáhlé. Pro lepší představu jsou v tabulce (Tabulka 3) uvedeny některé z parametrů [38].

Tabulka 3: Možnosti systému 800xA

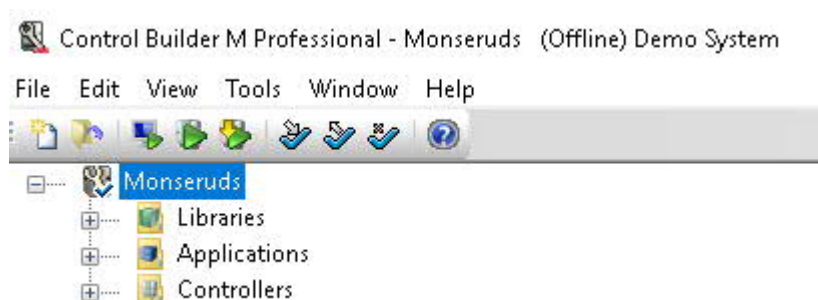
Celkový počet klientů (vzdálených i místních)	80
I/O kanály	Až 1000 na jeden kontrolér
Počet operátorských obrazovek	Až 160 na jeden systém
PROFIBUS konektivita	Až 2500 zařízení na 1 server
HART konektivita	Až 2500 zařízení na 1 server
PLC konektivita	Až 25000 signálů na 1 server

4 Implementace navrženého řešení do systému

Implementace řešení projektu je popsána tak, aby se podobné kroky neopakovali a naznačuje řešení různých situací. Celý program (ve formátu PDF i v podobě souborů pro obnovu projektu v systému 800xA) je přílohou této práce na samostatném nosiči. Samotný systém je programován z prostředí softwaru vyvinutého speciálně pro programování kontrolérů. V případě mé práce jde o program **Control Builder**. Tento program zajistí jednoduchost a rychlost v programování. Program umí pracovat s různými typy komunikačních protokolů a různými typy programovacích jazyků. Pracuje s programovacími jazyky, které jsou definované v normě **IEC61131-3** a komunikační protokoly **IEC60870-5** [44]. Aplikace, programy, kontroléry a další vytvořené soubory v programu Control Builder jsou synchronizovány se systémem 800xA, kde jsou vidět v jednotlivých strukturách (popsáno v kapitole 3.9.3) [43].

4.1 Vytvoření nového projektu

Po založení projektu (Kliknutím na File / New Project /AC 800M v Control Builderu) nás uvítá okno, ve kterém začneme pracovat (Obr. 14).



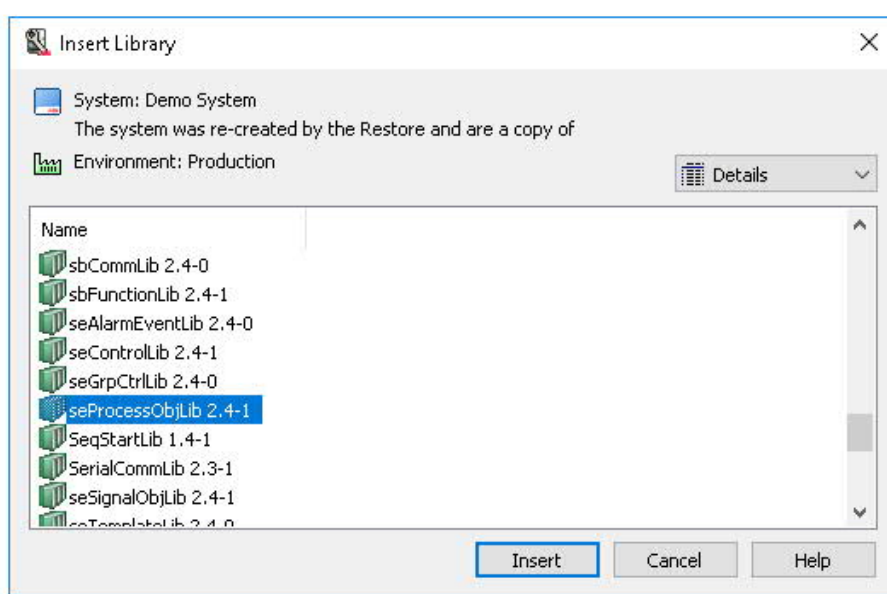
Obr. 14: Snímek obrazovky Control Builderu

V okně Libraries jsou přidávány softwarové knihovny obsahující funkční bloky potřebné k vytvoření programu nebo hardwarové knihovny zajišťující možnost přidávat komunikační moduly, I/O karty a další rozšíření. Program bude vytvářen v okně Applications a řídit bude kontroléry přiřazené v okně Controllers.

4.2 Přidávání knihoven do projektu

Knihovny jsou přidávány v okně Libraries. Nejprve je nutno přidat knihovnu obsahující programový blok potřebný k sestavení programu. Používání knihoven je výhodné z více důvodů. Jedním je možnost mít menší množství otestovaných programových bloků. Tyto bloky můžeme navíc používat i v budoucích aplikacích. Knihovny existují jak softwarové (zjednodušující programování), tak hardwarové (umožňující detekovat hardware [45]). Je nutné

zvolit si, jaké knihovny budeme potřebovat a k tomu nám poslouží dokumentace dostupná ke knihovnám. Nejprve je potřeba z funkčního popisu, z P&I diagramu a ze seznamu signálů projektu pochopit, jaké programové bloky hledáme. V tomto případě je určitě potřeba ovládat například ventily. Pro práci vybrány knihovny Utility Libraries 6.0, protože tato sada knihoven by měla obsahovat vše, co je potřeba k ovládní celku takového rozsahu. V dokumentaci k vybraným knihovnám ABB [46] je tedy například potřeba vyhledat takový blok, který by byl schopen ovládat ventil použitý v našem celku a přidat do projektu knihovnu, která tento blok obsahuje. K ovládní ventilu byla přidána knihovna seProcessObjLib (Obr. 15).



Obr. 15: Přidání knihovny do projektu

Pro práci se signály přidáme softwarovou knihovnu seSignalObjLib a tento postup je aplikován analogicky kdykoliv v průběhu projektu, když je zapotřebí využít bloku, který je obsažen v knihovně doposud nepřidané do projektu.

4.3 Nastavení hardwaru

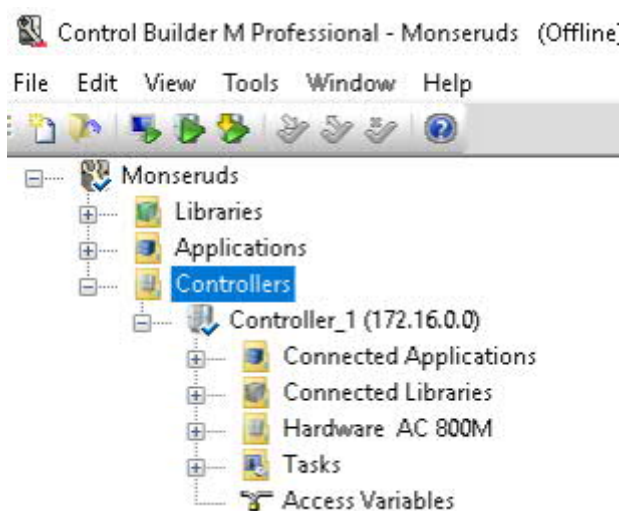
Po otevření okna controllers je zobrazena nabídka tak jak je na Obr. 16. V nabídce vidíme volby:

Connected Applications – Slouží pro přiřazení aplikace do kontroléru. V okně applications je možno vytvořit více programových aplikací a každé přiřadit jiný kontrolér.

Connected Libraries – Zde vybíráme, jaké z HW knihoven, které jsou již přidáné v projektu, budou propojeny s daným kontrolérem. V základní nabídce vybereme Controller_1 a nastavíme jeho IP adresu. V tomto případě bude testována aplikace na softwarovém kontroléru. Jde o náhradu za skutečný kontrolér sloužící simulaci, která se používá na testování předtím, než dojde k nahrání programu do skutečného zařízení.

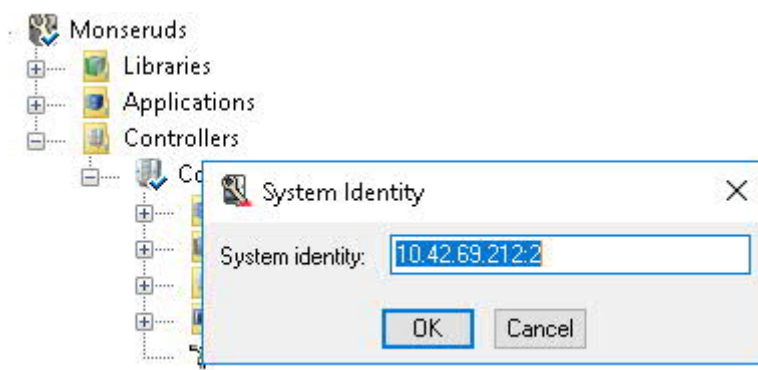
Hardware AC 800M – Zde se nastavují typy a adresy kontrolérů, vstupní a výstupní karty, komunikační periferie a další rozšíření.

Tasks – V jednom kontroléru může být nahráno více aplikací. Každá aplikace musí mít přiřazen svůj task, ten rozhoduje o rychlosti vykonání aplikace. Aplikace pro měření hladiny nebude potřebovat být vykonávána s tak rychlou odezvou jako například aplikace pro regulaci otáček motoru. V programu jsou tři přednastavené tasky – Slow(1000ms), Normal (250ms), Fast (50ms). V okně tasks můžeme upravovat parametry vytvořených „tasků“ nebo vytvářet nové „tasky“.



Obr. 16: Prostředí pro nastavení kontroléru

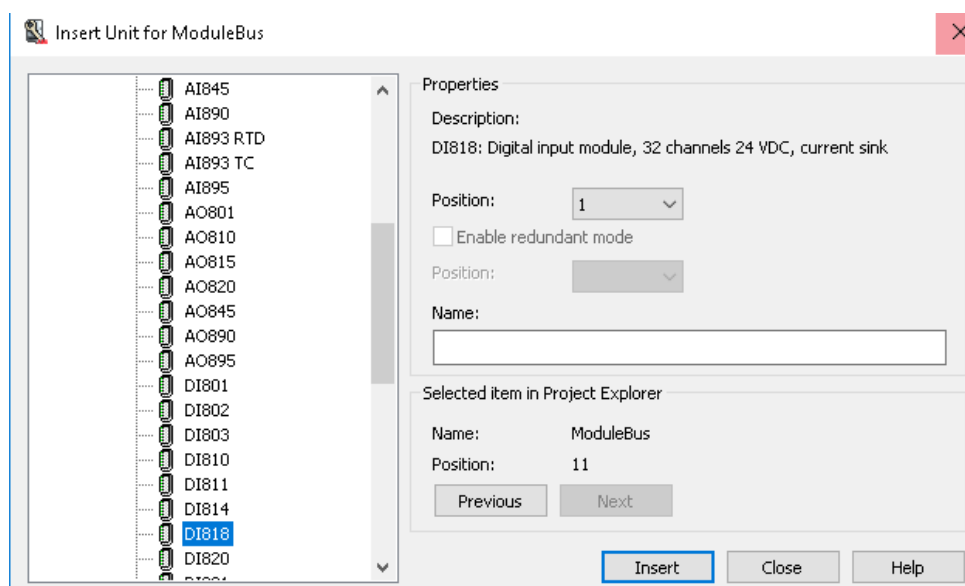
Nastavení adresy kontroléru – Po kliknutí pravým tlačítkem myši na **Controller_1** v okně Controllers zvolíme **Properties** a dále pak **System Identity**. Zobrazeno je okno pro nastavení adresy (Obr. 17). Zde se nastaví IP adresa počítače, na kterém běží aplikace softwarového kontroléru (popřípadě IP adresa skutečného kontroléru). Adresu zakončíme přívráskem :2 (odkazuje na softwarový kontrolér).



Obr. 17: Identita kontroléru

4.4 Přidávání HW modulů

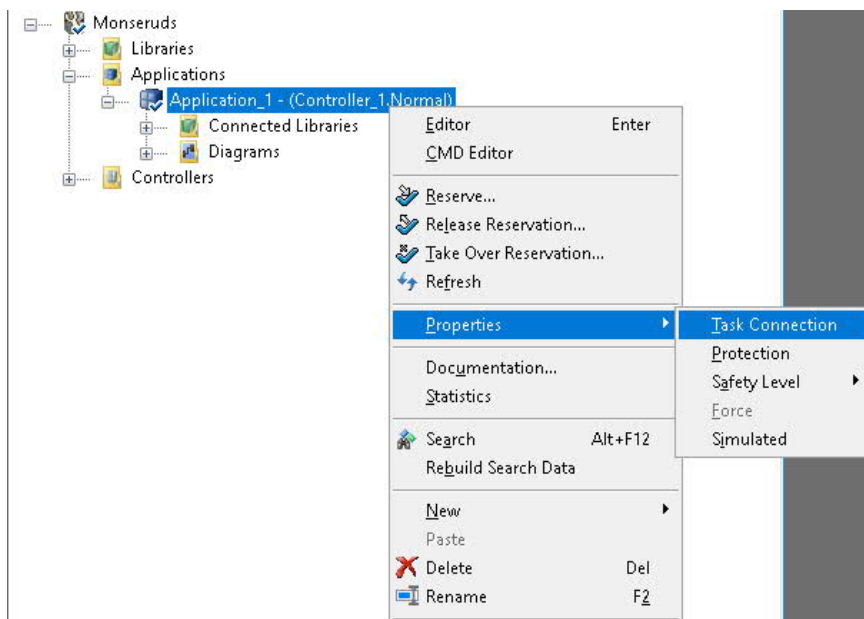
Potřeba je vybrat typ PLC použitý na projekt a přidat další rozšíření jako jsou vstupní a výstupní karty nebo komunikační moduly. Vstupní a výstupní karty lze přidat v podokně **Hardware AC 800M / PM866/TP830** (Vybraná procesorová jednotka tohoto projektu – lze změnit). Zde vybereme pravým tlačítkem myši **ModuleBus** (zajišťuje komunikaci s I/O kartami) a klikneme na InsertUnit (Obr. 18). V seznamu karet musí být vybrány karty, které jsou použity v projektu. Názvy v seznamu tedy odpovídají skutečným kartám. Přidané karty je možno zobrazit rozvinutím nabídky ModuleBus. Ke komunikaci s kartami bývá často používán také protokol profibus. V takovém případě jsou karty přidávány až po přidání této komunikace.



Obr. 18: Vkládání I/O modulů

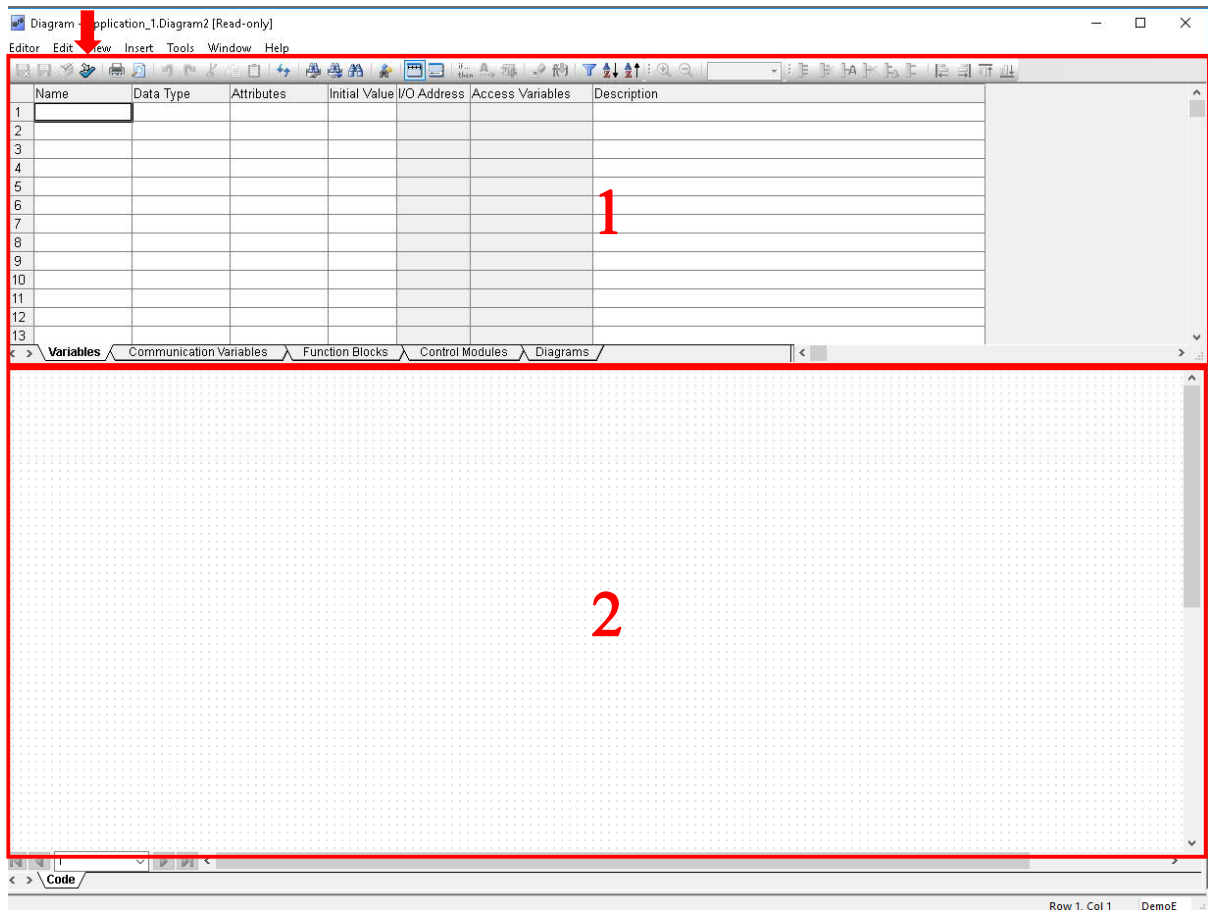
4.5 Postup programování

Program je vytvářen v podokně **Applications**. Po vytvoření nové aplikace (kliknutím pravým tlačítkem myši na Applications / New Application) je nutné přiřadit aplikaci task, ten je vytvořen v Controllers (Obr. 19). V tomto okně, stejně tak jako v okně Controllers je nutno připojit knihovny obsažené v projektu, které budou využity ve vytvořené aplikaci (prostřednictvím Connected Libraries).



Obr. 19: Přřazení Task Connection

V okně **Diagrams** byl analogickým způsobem vytvořen nový diagram. Každý vytvořený **diagram** je také provázán s „taskem“ z kontroléru. Po otevření diagramu je zobrazeno okno pro prohlížení a psaní programu (Obr. 20). Po prvním otevření je diagram v režimu pro čtení a nelze upravovat. Po kliknutí na ikonu „Reserve“ (označena červenou šipkou) má uživatel zarezervovaný diagram pro sebe a je schopen provádět úpravy. V okně označeném číslem 1 vytváří uživatel proměnné pro daný diagram (vytvořené proměnné lze vidět na Obr. 21). Globální proměnné vytváříme stejným způsobem, ale jsou zapisovány přímo do okna aplikace. Pro komunikaci mezi diagramy jsou využívány Communication Variables. Toho je využito například při měření průtoku UTL0-FT01. Tento signál je použit v aplikaci pro zachytávání písku, tak i v aplikaci pro zachytávání tuku. Okno 2 slouží pro vytváření programu.



Obr. 20: Funkční diagram

4.5.1 Alokace signálů

Fyzicky již mají HW signály svoji pozici navrženou dle principu popsaného v kapitole 3.8. Nyní je potřeba provést tento úkol (předat tuto informaci) i do prostředí programu Control builder. Na Obr. 21 je vytvořena například proměnná signálu UTL0_FT01_PQ1. Tento signál slouží pro měření průtoku. Průtok je zde měřen impulsním signálem, kde jeden puls představuje 400 l protečeného množství a informace o něm je binárním pulsem. Z tohoto důvodu byl pro proměnnou zvolen datový typ BoolIO. Zkratka IO znamená, že jde o proměnnou typu Input/Output. Takový typ má svoji reprezentaci v podobě skutečného HW signálu.

	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variables	Description
19	LTL2_PV01_GS1	BoolIO	retain				
20	LTL2_PV01_GS2	BoolIO	retain				
21	LTL2_FTD1_PQ1	BoolIO	retain				
22	LTL1_FTD1_PQ1	BoolIO	retain				
23	U05_DP01_GS1_	BoolIO	retain				
24	U05_DP01_GS2	BoolIO	retain				
25	UTLD_FTD1_PQ1	BoolIO	retain				
26							
27							

Obr. 21: Zápis proměnné

Dalším krokem je otevření I/O karty, na kterou má být signál přiřazen (Controllers / Hardware AC 800M / PM860/TP830 / ModuleBus). Tento digitální signál má být přiřazen na kartu DI818. Po otevření karty vidíme okno tak, jak je na Obr. 22. Ikonka Insert Path from Three (označena šipkou 1) otevře proměnné, které je možno přiřadit.

Channel	Name	Type	Variable	I/O Description
IXD.11.4.1	Input 1	BoolIO		
IXD.11.4.2	Input 2	BoolIO		
IXD.11.4.3	Input 3	BoolIO		
IXD.11.4.4	Input 4	BoolIO		
IXD.11.4.5	Input 5	BoolIO		
IXD.11.4.6	Input 6	BoolIO		
IXD.11.4.7	Input 7	BoolIO		

Obr. 22: Alokace signálu na kartu

Po správném přiřazení je v otevřeném okně dané karty vidět ve sloupci Variable cesta k signálu tak jak je to na Obr. 23.

Channel	Name	Type	Variable	I/O Description
IXD.11.2.20	Input 20	BoolIO	Application_1.Diagram4.LTL2_PV01_GS2	
IXD.11.2.21	Input 21	BoolIO	Application_1.Diagram4.LTL2_FT01_PQ1	
IXD.11.2.22	Input 22	BoolIO	Application_1.Diagram4.LTL1_FT01_PQ1	
IXD.11.2.23	Input 23	BoolIO	Application_1.Diagram4.U05_DP01_GS1_	
IXD.11.2.24	Input 24	BoolIO	Application_1.Diagram4.U05_DP01_GS2	
IXD.11.2.25	Input 25	BoolIO	Application_1.Diagram4.UTL0_FT01_PQ1	
IXD.11.2.26	Input 26	BoolIO		
IXD.11.2.27	Input 27	BoolIO		
IXD.11.2.28	Input 28	BoolIO		

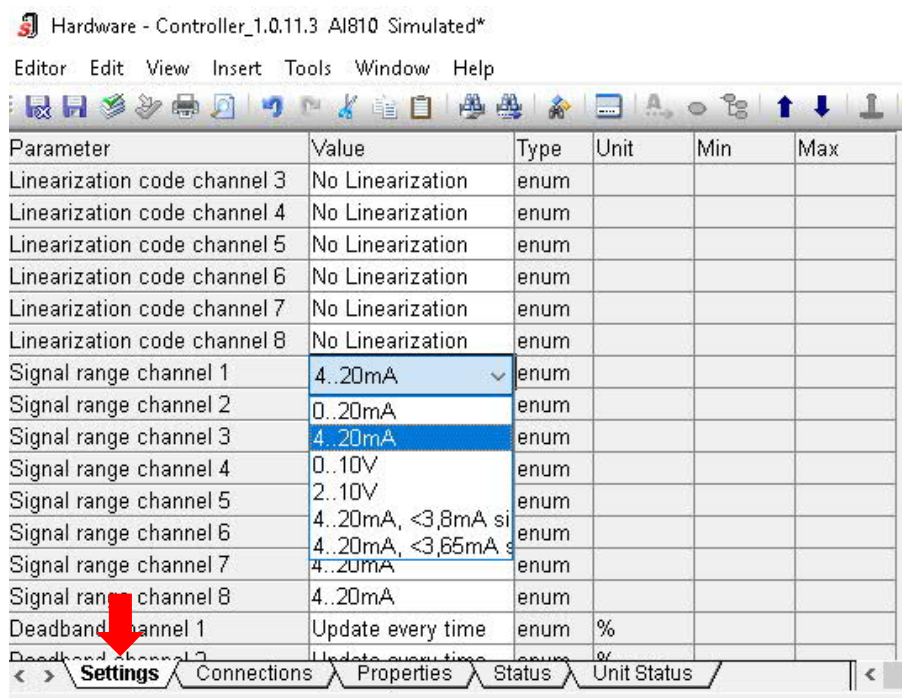
Obr. 23: Alokované signály

V případě alokace analogového signálu je nutné také nastavit rozsah tohoto signálu. Rozsah je nastavován v okně Properties (označeno šipkou 2). Změna rozsahu analogového signálu pro měření hladiny (FET_LT01_PV1) je vidět na Obr. 24. V základu program předpokládá rozsah 0–100. Ze zadání projektu – z listu signálů je ale patrné, že tento signál se bude pohybovat v rozmezí 0–5 m a proto je potřeba hodnotu maxima upravit (Obr. 24).

Channel	Variable	Min	Max	Unit	Fraction	Inverted
IWD.11.3.1	Application_1.Diagram4.FET1_LT01_PV1	0.0	5.0	m	1	false
IWD.11.3.2	Application_1.Diagram4.SAN1_SK01_WT1_PV1	-50.0	50.0	Kn	1	false
IWD.11.3.3		0.0	100.0		1	false
IWD.11.3.4		0.0	100.0		1	false
IWD.11.3.5		0.0	100.0		1	false
IWD.11.3.6		0.0	100.0		1	false

Obr. 24: Změna rozsahu signálu

V okně Settings nastavíme kanál podle toho, jakým způsobem je signál reprezentován – proudový nebo napěťový signál. V tomto projektu jsou analogové signály reprezentovány proudovou hodnotou v rozsahu 4–20 mA. Nastavení je vidět na Obr. 25.



Obr. 25: Nastavení typu signálu

4.5.2 Výběr vhodné metody programování

Program control builder umožňuje uživateli zvolit si režim programování. Režimy programování se kterými program pracuje jsou definovány normou IEC 61131-3.

Uživatel si vybírá mezi následujícími způsoby [47]:

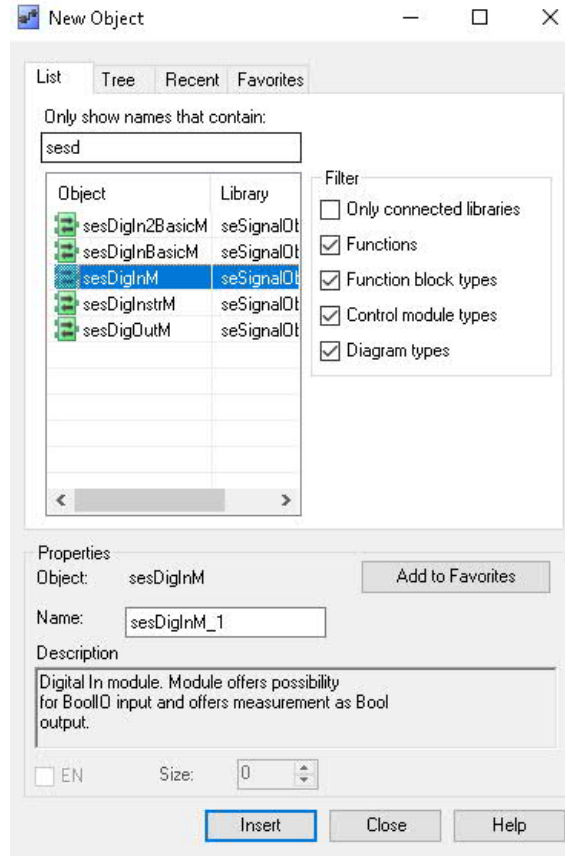
- IL – Instruction List – posloupnost instrukcí,
- ST – Structured Text – vyšší programovací jazyk – obdoba Pascalu,
- LD – Ladder Diagram – kontaktní plán / liniové či reléové schéma,
- SFC – Sequential Function – Chart vývojové schéma,
- FBD – Function Block Diagram – schéma funkčních bloků,
- CFC – Continuous Function Chart – volně propojované bloky. [47]

Každý z těchto režimů má svoje výhody i nevýhody. Programátor se při svoji práci nemusí omezit na jediný způsob, ale může pro svoji aplikaci použít kombinaci více režimů. Program tohoto projektu bude vytvářen za pomoci FBD diagramu.

4.5.3 Práce v diagramu funkčních bloků

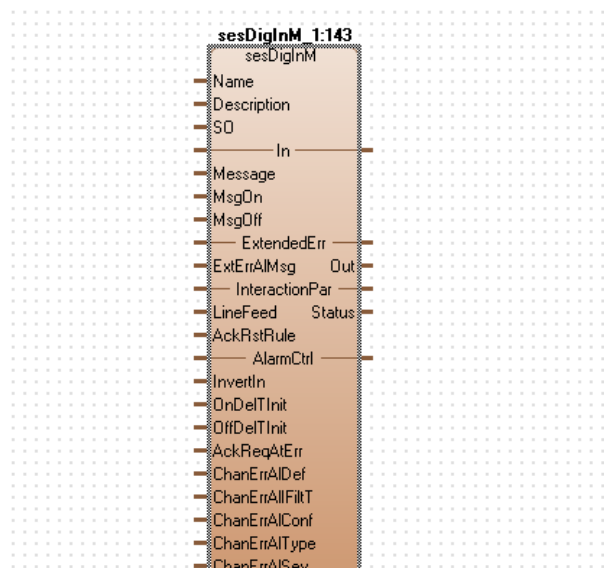
Práce začíná vložením vhodného bloku do diagramu. Například pro digitální vstup byl vybrán funkční blok sesDigInM. Tento blok je přidán kliknutím pravého tlačítka myši do otevřeného okna funkčního diagramu. Zvolením New/Object je otevřeno prostředí pro

přidávání funkčních bloků (Obr. 26). V okně jsou vidět pouze objekty obsažené v knihovnách přidávaných do projektu. V případě, že se hledaný objekt v okně nenalézá je potřeba přidat knihovnu s daným objektem do projektu prostřednictvím Libraries. Vybraný objekt je vložen volbou insert.



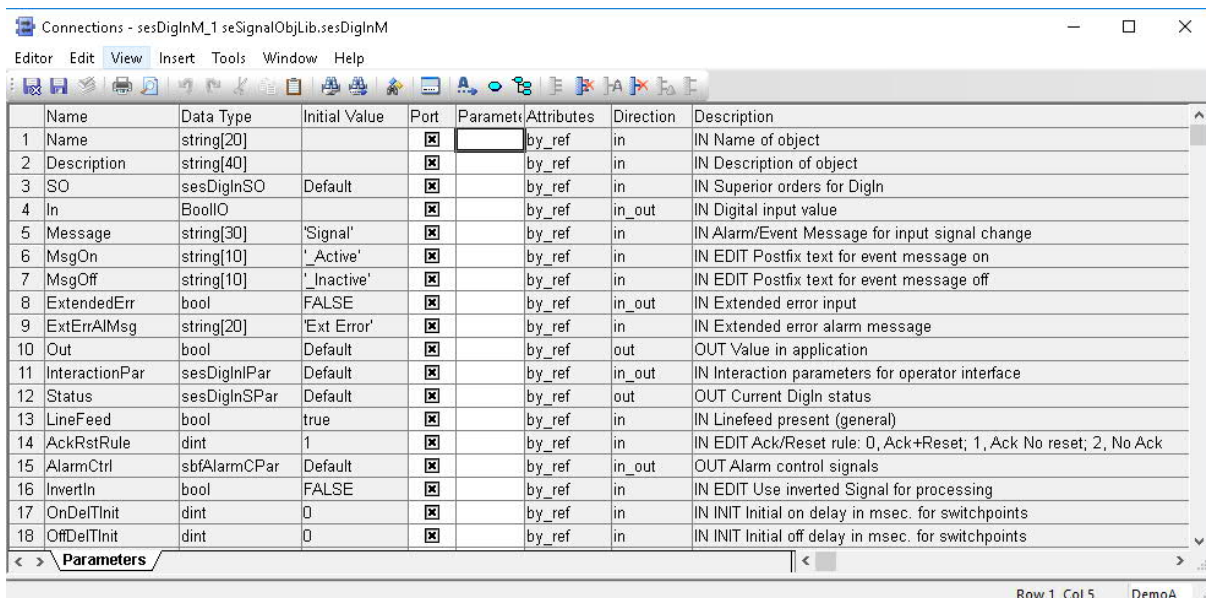
Obr. 26: Přidávání objektů do funkčního diagramu

Po vložení je blok zobrazen v poli funkčního diagramu (Obr. 27).



Obr. 27: Blok vložený do pole funkčního diagramu

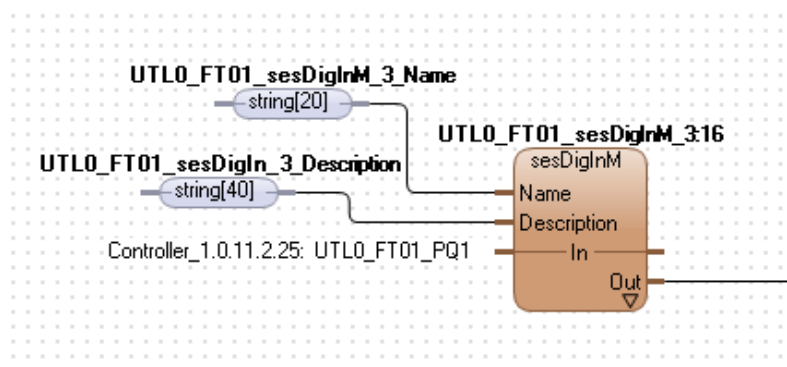
Některé z těchto vstupů je uživatel povinen obsadit, jiné záleží na potřebě programátora. Dvojklikem myši na blok jsou zobrazeny všechny vstupy a výstupy se svými parametry. Složitější bloky mají k dispozici opravdu velké množství parametrů tak jak je vidět na Obr. 28. Ve sloupci Description je popsán význam signálu. Na základě toho, zda je daný parametr potřeba použít je možné jeho port zapnout nebo vypnout křížkem ve sloupci Port.



Name	Data Type	Initial Value	Port	Parameters	Attributes	Direction	Description
1 Name	string[20]		<input checked="" type="checkbox"/>		by_ref	in	IN Name of object
2 Description	string[40]		<input checked="" type="checkbox"/>		by_ref	in	IN Description of object
3 SO	sesDigInSO	Default	<input checked="" type="checkbox"/>		by_ref	in	IN Superior orders for DigIn
4 In	BoolIO		<input checked="" type="checkbox"/>		by_ref	in_out	IN Digital input value
5 Message	string[30]	'Signal'	<input checked="" type="checkbox"/>		by_ref	in	IN Alarm/Event Message for input signal change
6 MsgOn	string[10]	'_Active'	<input checked="" type="checkbox"/>		by_ref	in	IN EDIT Postfix text for event message on
7 MsgOff	string[10]	'_Inactive'	<input checked="" type="checkbox"/>		by_ref	in	IN EDIT Postfix text for event message off
8 ExtendedErr	bool	FALSE	<input checked="" type="checkbox"/>		by_ref	in_out	IN Extended error input
9 ExtErrAlMsg	string[20]	'Ext Error'	<input checked="" type="checkbox"/>		by_ref	in	IN Extended error alarm message
10 Out	bool	Default	<input checked="" type="checkbox"/>		by_ref	out	OUT Value in application
11 InteractionPar	sesDigInPar	Default	<input checked="" type="checkbox"/>		by_ref	in_out	IN Interaction parameters for operator interface
12 Status	sesDigInSPar	Default	<input checked="" type="checkbox"/>		by_ref	out	OUT Current DigIn status
13 LineFeed	bool	true	<input checked="" type="checkbox"/>		by_ref	in	IN Linefeed present (general)
14 AckRstRule	dint	1	<input checked="" type="checkbox"/>		by_ref	in	IN EDIT Ack/Reset rule: 0, Ack+Reset; 1, Ack No reset; 2, No Ack
15 AlarmCtrl	sbfAlarmCPar	Default	<input checked="" type="checkbox"/>		by_ref	in_out	OUT Alarm control signals
16 InvertIn	bool	FALSE	<input checked="" type="checkbox"/>		by_ref	in	IN EDIT Use inverted Signal for processing
17 OnDelTInit	dint	0	<input checked="" type="checkbox"/>		by_ref	in	IN INIT Initial on delay in msec. for switchpoints
18 OffDelTInit	dint	0	<input checked="" type="checkbox"/>		by_ref	in	IN INIT Initial off delay in msec. for switchpoints

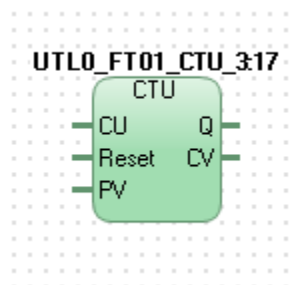
Obr. 28: Parametry bloku sesDigInM

K bločku je dostupná interní ABB dokumentace ([48]). Mezi povinné parametry patří Name a Description. Jméno samotného bločku lze změnit pouhým poklepáním myši na jeho název. Binární proměnná UTLO_FT01_PQ1 je již vytvořena, nyní je připojena do portu In v bločku digitálního vstupu sesDigInM. Po vypnutí pro projekt nepotřebných portů vypadá bloček tak jak je na Obr. 29.



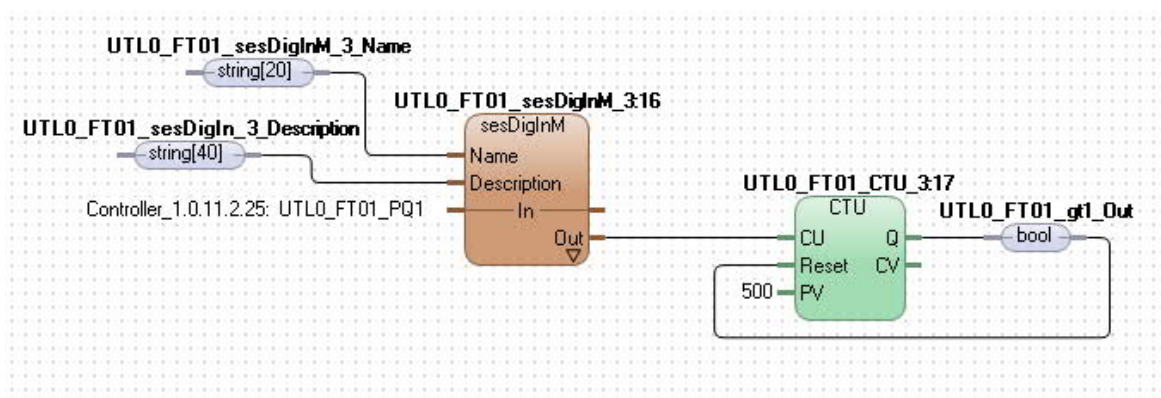
Obr. 29: Bloček obsluhující digitální signál

Tento pulsní signál reprezentuje průtok. Po průtoku 400 l je generován jeden puls. Zařazením čítače je umožněno čítání těchto pulsů. Bloček čítače (Obr. 30) je k nalezení v základní knihovně BasicLib.



Obr. 30: Bloček čítače

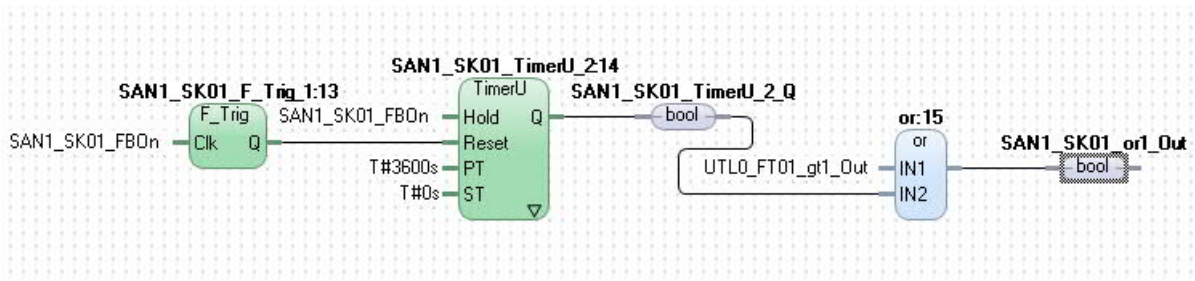
Tento čítač je velice jednoduchý. Na vstup CU je přiváděn binární signál, který chceme přičítat. Poté co je dosaženo hodnoty přivedené na vstup PV, je přiveden binární výstup Q do logické 1. Čítač pak počítá dál až do doby, než je přiveden signál na vstup Reset. Po přivedení signálu na vstup Reset je čítač vynulován a jeho výstup Q přejde do nuly. Výstup CV (counter value) je datového typu dint a nese informaci o počtu napočítaných pulsů. V tomto projektu je potřeba detekovat 200 m³ (pro sestavení programu splňující požadavky diagramu na Obr. 6). Je tedy potřeba, aby čítač napočítal 500 pulsů (1 puls = 400 l, 500 pulsů = 200 000 l). Měření průtoku je tedy realizováno za pomoci programu na Obr. 31. Vlevo na obrázku vstupuje signál do diagramu, vpravo jsou počítány pulsy. Proměnná UTLO_FT01_gt1_Out (vytvořena pravým kliknutím myši do pole diagramu New/Variable) je v logické 1 je-li aktivován výstup čítače a zároveň vlastní čítač vyresetuje.



Obr. 31: Měření průtoku

Výstup Q je tedy aktivní jen krátkou dobu, nicméně její krátký puls je využit v další části programu, kde aktivuje pohon škrabáku písku usazeného na dně nádrže. Z diagramu na Obr. 6 je ale patrné, že škrabák má být aktivován i v případě, že nebyl spuštěn za posledních 60 min.

Pro realizaci této podmínky je logicky vybrán časovač. Nová podmínka je vytvářena na nové stránce tohoto diagramu (Pravý klik myši / Pages/ Add New Page). Realizace této podmínky je vidět na Obr. 32. Tento typ časovače měří čas nahoru není-li aktivní binární vstup Hold, ten čítání pozastaví. V tomto případě je do tohoto vstupu přivedena zpětná vazba z pohonu škrabáku písku. Aktivní je jen v případě, že je spuštěn pohon škrabáku.

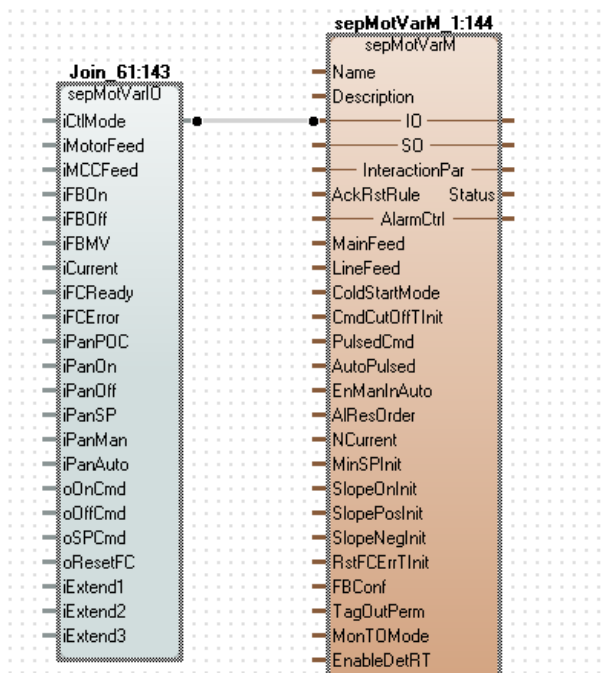


Obr. 32: Nastavení časovače

Vstup Reset vynuluje čítač. Je do něj přivedena sestupná hrana signálu zpětné vazby pohonu škrabáku. Sestupnou hranu detekuje bloček F_Trig. Tento bloček na svém vstupu očekává binární signál. Přejde-li tento signál ze stavu logické 1 do 0, je na výstup přiveden puls [49]. Měření času je tedy vynulováno vždy ve chvíli, když je motor vypnut. Nenastane-li takový moment za dobu 60 minut, je výstup časovače aktivován. Výstup časovače je naveden do bloku OR, kde se setkává s druhou podmínkou pro rozběh pohonu (měření průtoku). Proměnná SAN1_SK01_or1_Out bude tedy v logické 1 pouze v případě, že bude splněna alespoň jedna z podmínek pro rozběh motoru. Proměnná svým impulsem spouští pohon.

4.5.4 Strukturovaný datový typ

Motor je ze zadání ovládán frekvenčním měničem a škrabák má být tažen vpřed jinou rychlostí než při svém návratu. Na práci tohoto typu je vytvořen blok pro ovládání motoru s proměnlivou rychlostí – sepMotVarM [50]. Tento blok je již složitější a jeho obsluha vyžaduje seznámení se speciálními proměnnými knihovny Utility Library (sepMotVarIO, sepMotVarSO, sepMotVarIPar, sbfAlarmCPar a další). Za těmito proměnnými je schováno více dalších proměnných – je to tedy strukturovaný datový typ [49]. K jejich rozkladu lze využít funkce **Join** a **Split**. Funkce Join datový typ spojuje, Split naopak rozděluje. Na Obr. 33 je vidět použití funkce Join na parametr sepMotVarIO.

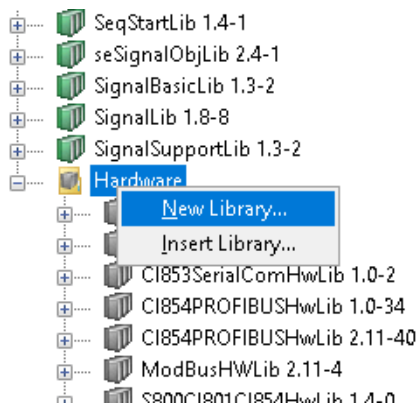


Obr. 33: Použití funkce Join

Parametr sepMotVarIO slouží ke komunikaci s vnějšími I/O periferiemi. V případě tohoto projektu, je motor ovládán frekvenčním měničem. Mezi frekvenčním měničem a PLC je zajištěna komunikace PROFIBUSEM.

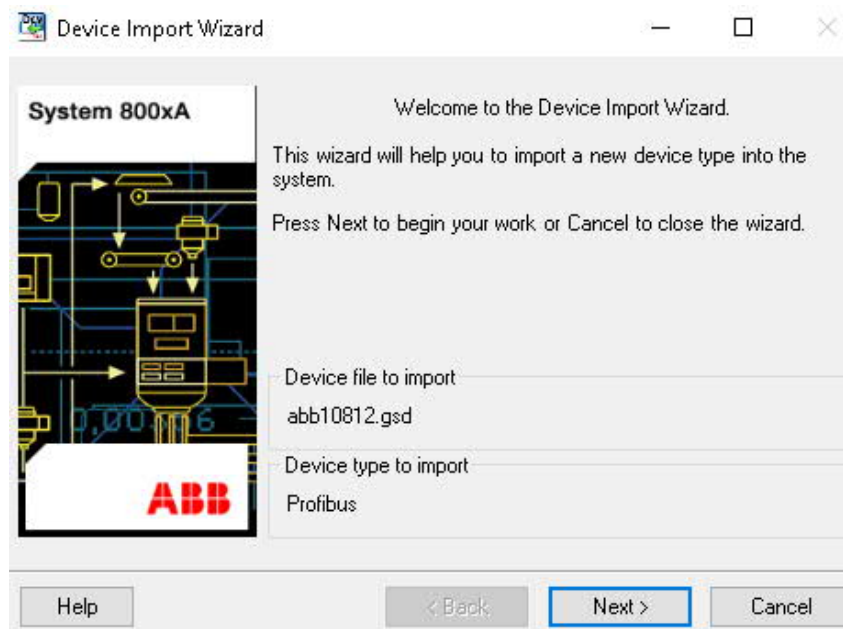
4.5.5 Implementace PROFIBUS komunikace

Pro implementaci zařízení, které komunikuje přes PROFIBUS jsou vytvořeny takzvané GSD (General Station Description) soubory. Tento soubor popisuje zařízení a umožňuje jeho integraci do systému. Často je nazýván elektronickým datasheetem [51]. K nalezení jsou takové soubory přímo na stránce www.profibus.com, popřípadě na stránce výrobce daného zařízení. V control builderu je obsažen konfigurační nástroj (**Device Import Wizard**) pro práci s GSD soubory. Ve volbě pro přidávání knihoven (volba Libraries z kapitoly 4.2) je vytvořena nová hardwarová knihovna (Obr. 34).



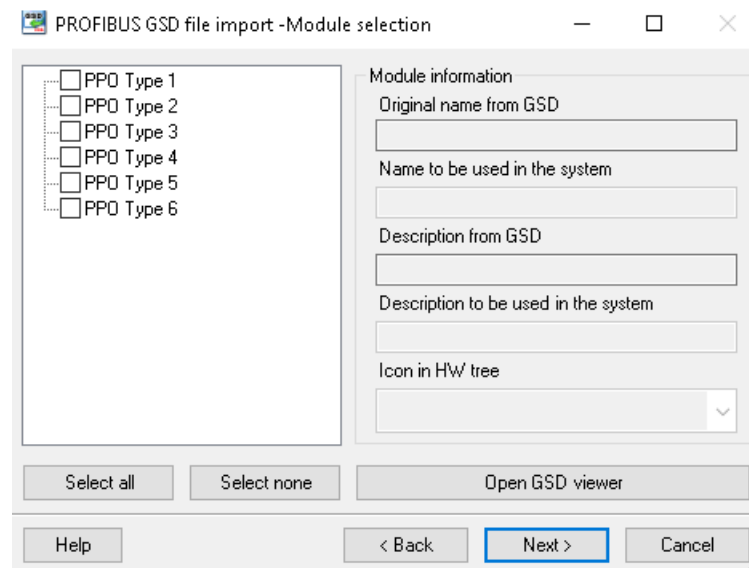
Obr. 34: Vytvoření nové knihovny

Po přidání je knihovna prázdná. Pravým klikem myši na knihovnu přes výběr insert je vybrána cesta k GSD souboru (je tedy potřeba v tuto chvíli již mít soubor stažený). Tím je spuštěn Device Import Wizard (Obr. 35). V tomto projektu je použit PROFIBUS adaptér RPBA-01 a GSD soubor tak patří k tomuto adaptéru.



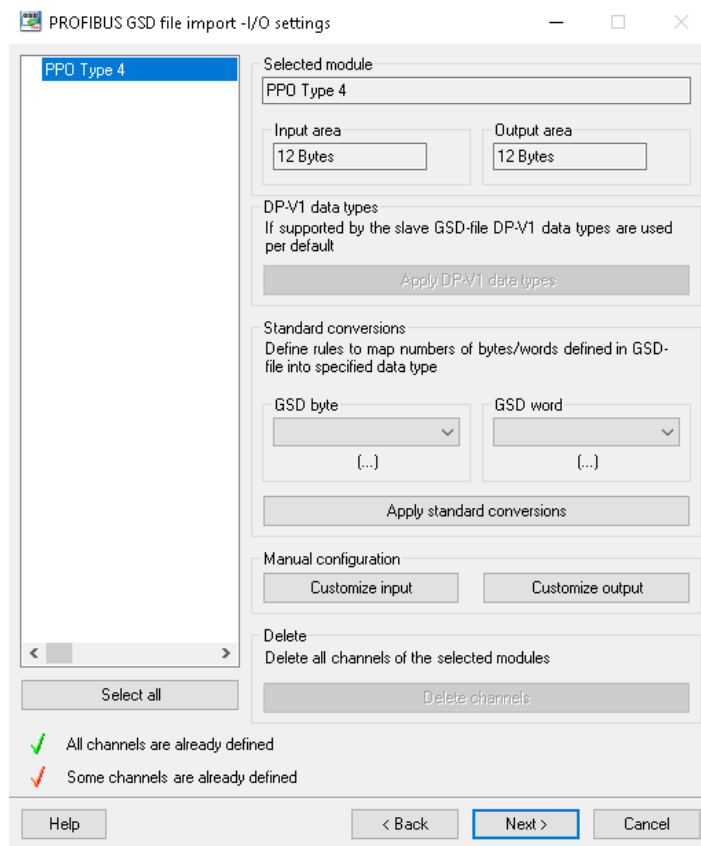
Obr. 35: Device Import Wizard

Pokračování zavede uživatele na volbu typu telegramu (Obr. 36). Profibus posílá informace ve formě telegramu. Typ telegramu je zvolen podle specifikace použitého profibus adaptéru [52]. V závislosti na zvoleném telegramu se mění jeho délka a počet bitů, které může uživatel nastavit [53]. Podle toho, kolik proměnných potřebuje uživatel obsadit, vybírá daný typ telegramu. Telegram může nabývat až 256 bytů, z toho 11 bitů nese informaci, ale jsou použity na režii telegramu. Říkáme, že jde o hlavičku telegramu [53].



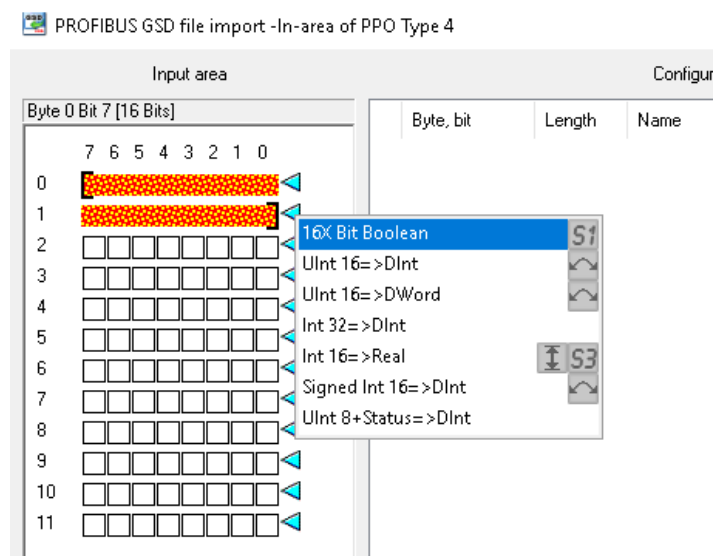
Obr. 36: Volba typu telegramu

Pro tento projekt byl vybrán telegram PPO Type 4, protože není potřeba vytvářet vlastní speciální proměnné na jednotlivé bity a tento telegram obsahuje již všechny potřebné povely. Dalším úkolem je nadefinovat v konfiguračním prostředí datové typy jednotlivých bytů. Tato definice je prováděna zvlášť pro vstup a zvlášť pro výstup (Customize input, Customize Output na Obr. 37). Při definování datového typu bytů je uživatel odkázán na dokumentaci k zařízení, které využívá. V tomto případě PROFIBUS adaptér RPBA-01 [52].



Obr. 37: Konfigurace profibusu

Z dokumentace ([52]) vyplývá, že první dva byty jsou fixní a je nutné je připravit pro Control Word a Referenci (v případě definování výstupu) nebo Status Word a Actual Value (v případě definování vstupu). Tato fixní oblast bude nadefinována jako 16x bit Boolean. Definování je provedeno označením daného bytu a zvolením datového typu (Obr. 38). Oblast následujících bytů je pro uživatele volitelně definovatelná.



Obr. 38: Konfigurace PROFIBUS komunikace

PROFIBUS je nyní nadefinován v nově vytvořené knihovně. Tato knihovna je přidána do projektu podle postupu z kapitoly 4.2. Nyní lze v okně Hardware vedle přidané jednotky PLC vidět také PROFIBUS (Obr. 39). Komunikační rozhraní CI854 je PROFIBUS master modulem komunikujícím za pomoci standardu sériové komunikace RS-485 [54].



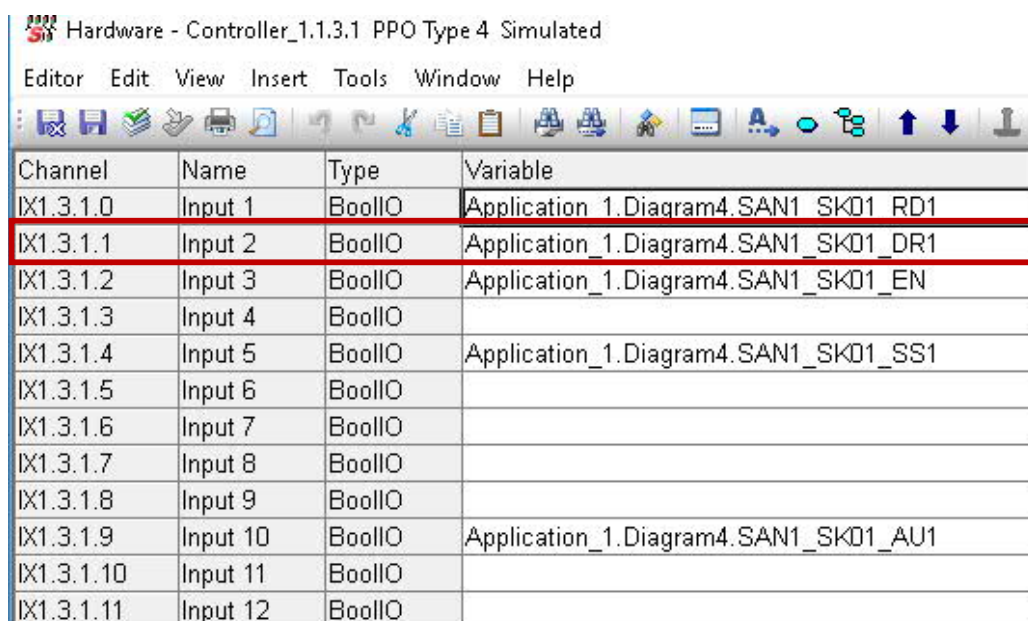
Obr. 39: Adaptér RPBA-01

PPO Type 4 je typ zvoleného telegramu. Podobně jako při alokaci kanálů na I/O karty, je nutné alokovat signály na telegram. V dokumentaci k použitému adaptéru RPBA-01 ([52]), je potřeba vyčíst, jaké bity Status Wordu a Control Wordu jsou připraveny na proměnné, které je potřeba použít. Pro ukázkou je proces alokace proveden na příkladu parametru RDY_RUN (READY TO OPERATE). Podle dokumentace k adaptéru RPBA-01[52] (Obr. 40) je vidět, že bit 1 má být obsazen parametrem, který nese informaci Ready to operate.

Bit	Name	Value	STATE/Description
0	RDY_ON	1	READY TO SWITCH ON
		0	NOT READY TO SWITCH ON
1	RDY_RUN	1	READY TO OPERATE
		0	OFF1 ACTIVE
2	RDY_REF	1	ENABLE OPERATION
		0	DISABLE OPERATION
3	TRIPPED	1	FAULT
		0	No fault
4	OFF_2_STA	1	OFF2 inactive
		0	OFF2 ACTIVE
5	OFF_3_STA	1	OFF3 inactive
		0	OFF3 ACTIVE

Obr. 40: Některé parametry Status Wordu

Nyní je potřeba provést alokaci v prostředí Control Builderu. Dvojklikem myši na telegram je otevřeno okno pro alokaci signálů (Obr. 41). V listu se signály (I/O list zadávací dokumentace) je vyhledán signál, který nese tuto informaci a je alokován na bit 1.

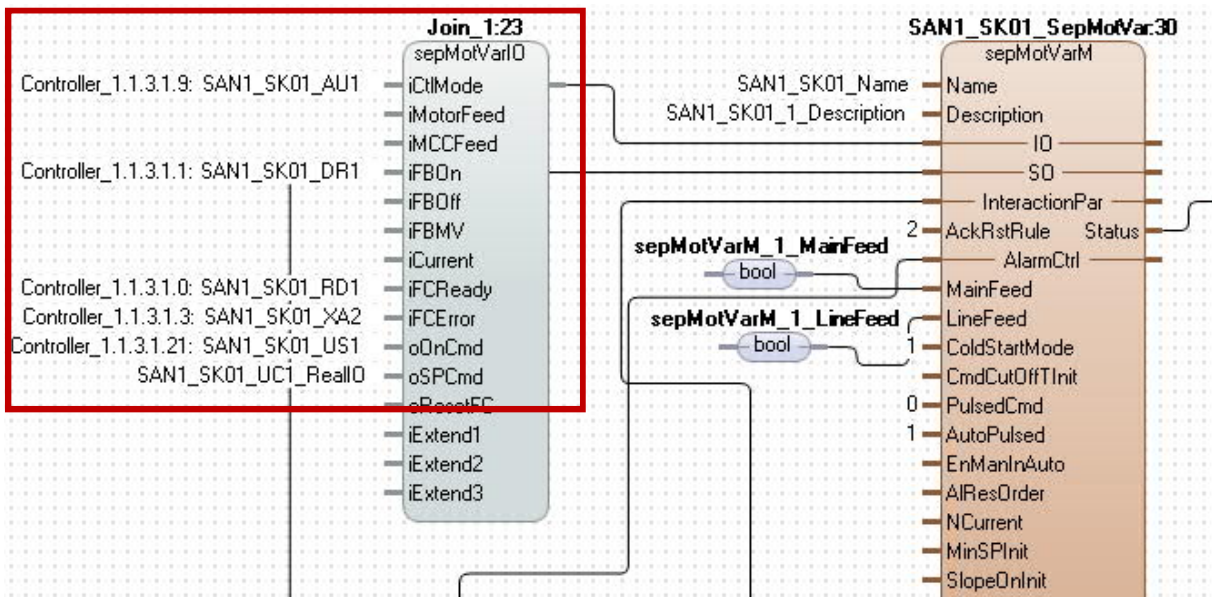


Channel	Name	Type	Variable
IX1.3.1.0	Input 1	BoolIO	Application_1.Diagram4.SAN1_SKD1_RD1
IX1.3.1.1	Input 2	BoolIO	Application_1.Diagram4.SAN1_SKD1_DR1
IX1.3.1.2	Input 3	BoolIO	Application_1.Diagram4.SAN1_SKD1_EN
IX1.3.1.3	Input 4	BoolIO	
IX1.3.1.4	Input 5	BoolIO	Application_1.Diagram4.SAN1_SKD1_SS1
IX1.3.1.5	Input 6	BoolIO	
IX1.3.1.6	Input 7	BoolIO	
IX1.3.1.7	Input 8	BoolIO	
IX1.3.1.8	Input 9	BoolIO	
IX1.3.1.9	Input 10	BoolIO	Application_1.Diagram4.SAN1_SKD1_AU1
IX1.3.1.10	Input 11	BoolIO	
IX1.3.1.11	Input 12	BoolIO	

Obr. 41: Alokace signálů na PROFIBUS

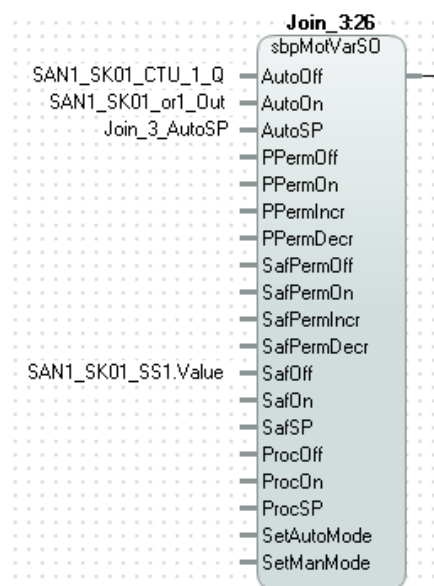
4.5.6 Ovládání motorového bloku

V této chvíli, je možno pokračovat s prací započatou ve funkčním diagramu na Obr. 33. V diagramu lze nyní již navázat signály na blok ovládání motoru (Obr 42). Proměnné v červeném rámečku komunikují přes profibus s frekvenčním měničem. Tyto proměnné jsou typu RealIO a reprezentují tedy HW signál. Přímo na bločku motoru je nastaven charakter povelů, způsob schvalování alarmů, výchozí režim, ve kterém se motor nachází po startu systému (automatický nebo manuální), počet feedbacků a další parametry. Například tento motor dostává z logiky pulsní signály (AutoPulsed=1), ale povely, které jdou na měnič mají kontinuální úroveň (PulseCmd=0). Po startu systému je motor v automatickém režimu (ColdStartMode=1). Motor má pouze 1 feedback a to feedback on (FBConf=1).



Obr 42: Profibus signály navázané na bloček motoru

Signály, které přicházejí z logiky (z programu) ovládají bloček přes parametr SO – Superior Order (Obr. 43).



Obr. 43: Ovládání motorového bloku z logiky

V logice jsou využívány také zpětné vazby, tedy informace o stavu, ve kterém se motor nachází. Tyto informace jsou obsaženy v parametru Status. Použitím funkcí join a split jsou dále rozkládány i další strukturované proměnné a jsou postupně navazovány signály pro ovládání motoru a také zpětné vazby z motoru. Obdobně byl nastaven ventil. Pro tento projekt byl použit solenoidní ventil. Ventil má svoji výchozí pozici – normálně closed a otevřen je signálem US1 pouze po dobu, po kterou je tento signál aktivní. Dle typu ventilu byl tedy opět nastaven jeho ovládací blok.

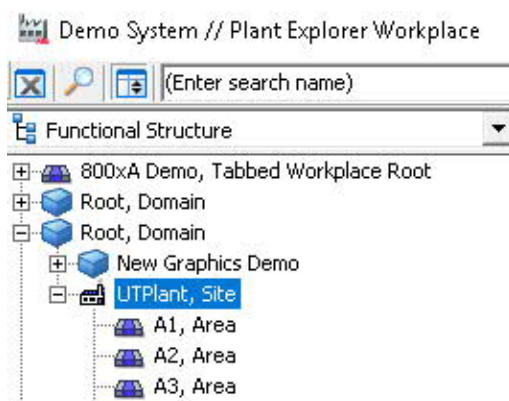
4.5.7 Možnosti programu Control Builder

V control builderu je možno výše uvedenými postupy vytvořit až [43]:

- Maximálně 64 programů v jedné aplikaci
- Maximálně 256 aplikací na jeden projekt v kontrolní struktuře
- Maximálně 8 aplikací na jeden kontrolér
- Maximálně 32 „tasků“ na jeden kontrolér
- Maximálně 32 kontrolérů na jeden projekt v kontrolní struktuře [43]

5 Tvorba grafického rozhraní

Grafické rozhraní pro operátora je v tomto projektu vytvářeno v prostředí Graphics Builderu. V kapitole 3.9.2 je popsána provázanost objektu systémem 800xA a je tak možno lépe pochopit provázanost grafického objektu s programem logiky. Při tvorbě grafického operátorského rozhraní je tento koncept velice výhodný, protože již při přidání funkčního bloku obsahující faceplate do diagramu v prostředí Control Builderu, je vytvořen ve struktuře systému aspekt s grafickým elementem [49]. Graphics Builder obsahuje knihovnu objektů a tyto objekty vykresluje vektorově. Uživatel si tak může přizpůsobit velikost objektu aniž by došlo ke ztrátě kvality zobrazení objektu [55]. Ve struktuře systému mají operátorské grafiky svoje místo ve **Functional Structure**. Systémem je v této struktuře předpřipraveno několik oblastí pro tvorbu grafik. Operátorské grafiky jsou zde rozčleněny podle technologie a umístění zařízení do jednotlivých oblastí. Oblast je nazývána jako Area (Obr. 44).



Obr. 44: Oblasti pro vytváření grafik

Area obsahuje více aspektů. Jedním z těchto aspektů je Area Display. Tento aspekt je souborem, pro editaci v programu Graphics Builder (Obr. 45).

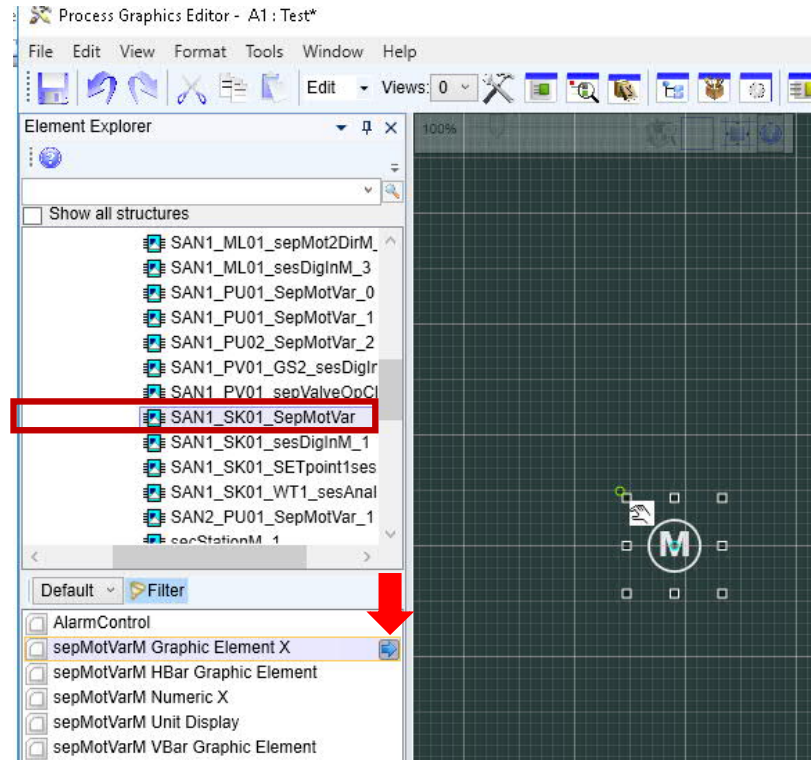
Aspects of 'A1'	Modified	Description	Inherited	Category name
.Alarm List	3/14/2008 1:42:4...	This aspect ca...	False	Alarm and Event List
.Event List	3/14/2008 1:42:4...	This aspect ca...	False	Alarm and Event List
Area Display Link	8/27/2014 8:45:2...	Graphic Eleme...	False	Graphic Element PG2
Area Display PG2	10/21/2010 11:2...	Graphic displa...	False	UT Graphic Display PG2
Area Navigation Element	4/21/2009 4:37:3...	Graphic Eleme...	False	Navigation Element PG2
Area Operator Trend	12/12/2008 4:48:...	Operator Tren...	False	Operator Trend
Area Trend Display	12/12/2008 4:48:...	Trend Display ...	False	Trend Display

Obr. 45: Aspekt grafického displeje

5.1 Grafické objekty

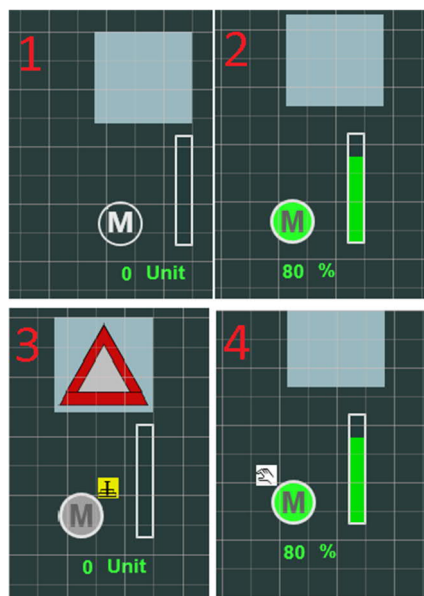
Grafický aspekt displeje je otevřen pro editace přes pravý klik myši a následně volbu Edit. Na následujícím obrázku (Obr. 46) je vidět jakým způsobem je objekt vkládán v prostředí

Graphics Builderu do oblasti displeje. Graphic Builder umí také procházet strukturami. Pro ukázkou je v kontrolní struktuře označen motorový blok SAN1_SK01. Za pomoci Element Exploreru (součást Graphics builderu) je vložen grafický element motoru do displeje (Označeno červenou šipkou v obrázku).



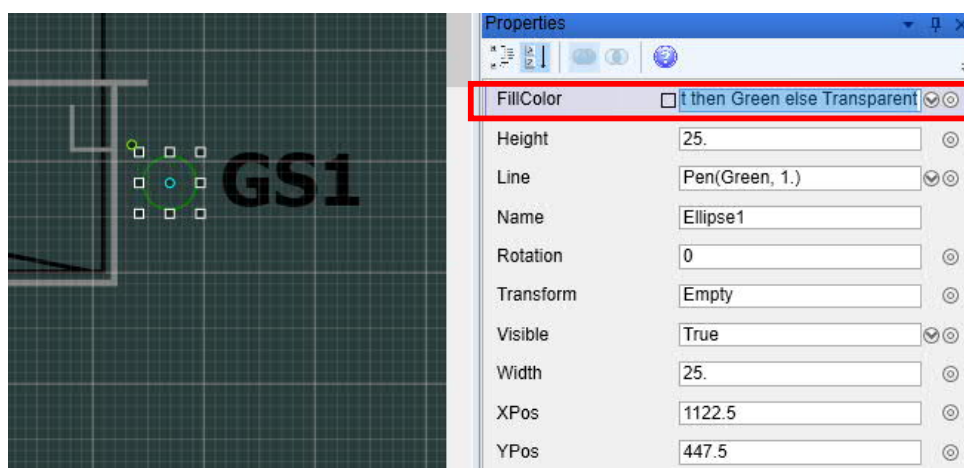
Obr. 46: Grafický objekt vložený do prostředí Graphics Builderu

K motoru je možné přidat více ukazatelů, reprezentující různý stav. Tyto objekty dynamické a jejich podoba se mění v závislosti na stavu, ve kterém se nachází skutečný objekt [55]. Příklady různých stavů pro objekt motoru je možno vidět na Obr. 47. V případě 1 motor neběží, případ 2 symbolizuje běh motoru na 80 %, v případě 3 generuje motor alarm (různých typů alarmů je mnoho) a v případě 4 běží motor na 80 % ale hodnota je nastavena manuálně operátorem a nepřichází z logiky.



Obr. 47: Různé stavy motoru

Mimo dynamických objektů jsou kresleny v Graphics Builderu také statické. Statické objekty svoji podobu nemění, ale jejich používání pomáhá operátorovi pochopit umístění sledovaného objektu v reálném systému. Graphics builder obsahuje spoustu knihoven s různými objekty. Ze statického objektu je také možno udělat objekt dynamický. Toho bylo využito například při průchodu škrabáku domácí pozicí. Na Obr. 48 je vidět původně statické průhledné kolečko. Barva tohoto kolečka je však podmíněna jednoduchou podmínkou (if SAN1_SK01_sesDigInM_1:Out then Green else Transparent) a při průchodu škrabáku domácí pozicí je o tom operátor informován zelenou barvou uvnitř tohoto kolečka.



Obr. 48: Programování podmínek v grafickém objektu

V případě proudové ochrany byl použit podobný způsob indikace. Zde jsou ale indikovány dvě úrovně alarmů (High Alarm a High High alarm), podmínka byla naprogramována tak, aby při indikaci High Alarmu byla zobrazena žlutá barva, zatímco High High alarm rozsvítí červenou barvu (if Diagram4: Split_5_HLevAct && !Diagram4:Split_5_HHLevAct then

Yellow else if Diagram4:Split_5_HHLevAct then Red else Transparent). Vykřičník v podmínce představuje negaci a ošetřuje situaci, kdy jsou aktivní oba alarmy a to tím způsobem, že je splněna jen podmínka pro barvu vyššího alarmu. Indikátor ve stavu High High alarmu je vidět na Obr. 49.



Obr. 49: Indikován High High alarm (přetížení na ochraně WT1)

5.2 Filozofie barev

Barvy objektů mohou být pevně nastaveny uživatelem, ale je vhodné držet se jistě v oboru očekávané filozofie. V dodržování pevně dané filozofie je systém 800xA vhodným pomocníkem. Jedna ze struktur systému – Workplace strukture umožňuje definovat barvy alarmů dle jejich důležitosti. Důležitost alarmu je v systému označována anglickým pojmem severity. Vybraná filozofie barev považuje červenou barvu za alarm vysoké priority, žluté barvy mají nižší prioritu a zařízení, které je v provozu svítí zelenou barvou. Alarmy jsou nastaveny tak, aby operátor viděl vždy barvu nejdůležitějšího neschváleného alarmu (Operátor může alarm schválit a potvrdit tím, že je sním seznámen). Alarm po jeho schválení má šedou barvu uvnitř červeného trojúhelníku výstrahy. Všechny schválené alarmy jsou zaznamenány a operátor může nahlížet na to, jaké alarmy byli schváleny. Symboly, barvy symbolů, barvy alarmů a další dodržují normami stanovené principy. Obecně se i u konkurenčních systémů (např. Siemens Tia Portal) dodržuje stejný standard. V tomto standardu červená barva znamená nebezpečí nebo stop stav. Zelená barva signalizuje zařízení v provozu nebo v bezpečném stavu. Žlutá je varovným symbolem. Pozadí je voleno tak, aby bylo co nejméně rušivým elementem a obvykle může být šedivé, popřípadě tmavě zelené barvy. Zvolený font musí být v kontrastu s pozadím [58]. Systém je vytvořen v souladu s normou IEC 60073 o základních a bezpečnostních principech pro interface mezi člověkem a strojem [56]. Dodržování standardů při tvorbě grafik má mimo jiné za cíl:

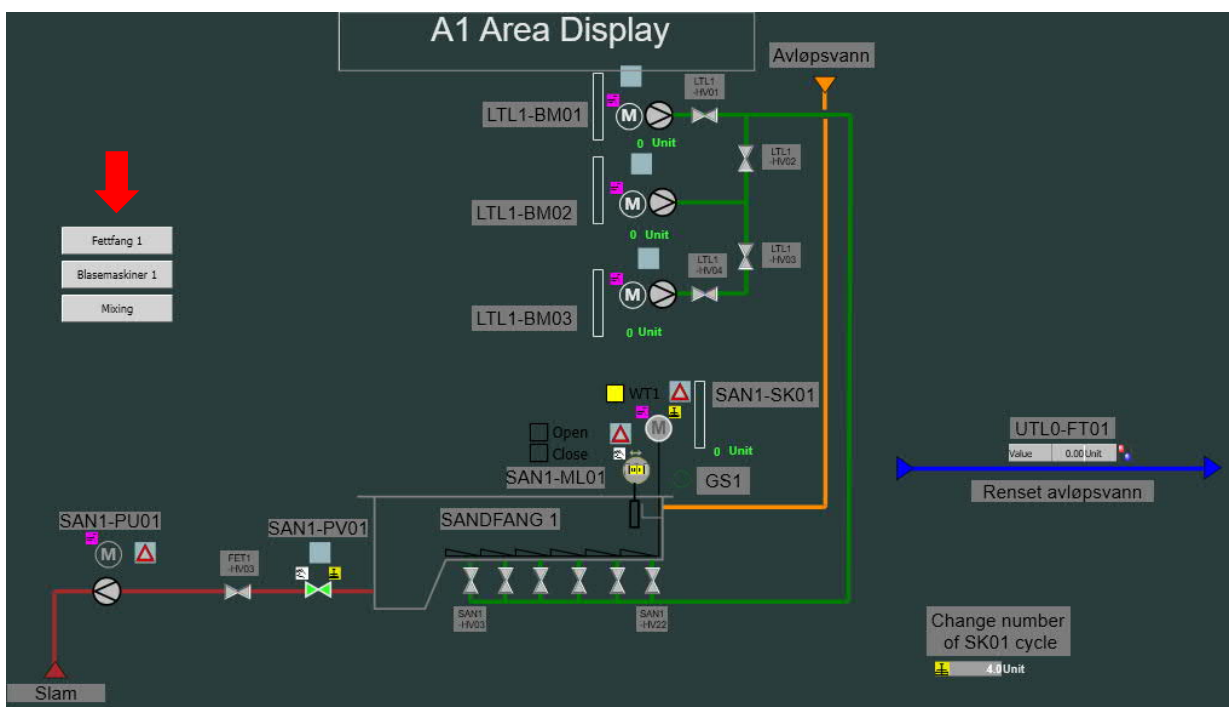
- zvýšit bezpečnost osob, majetku a / nebo životního prostředí;
- monitorování a řízení zařízení nebo procesu;
- usnadnit rychlé rozpoznání řídicích podmínek a poloh akčního členu [57].

Standardů, dle kterých rozhraní člověk-stroj vzniklo je ale více [56]:

- IEC 60417 Grafické značky pro použití na předmětech;
- IEC 60447 Základní a bezpečnostní zásady pro rozhraní člověk-stroj;
- IEC 60617 Grafické značky pro schémata (Již neplatná);
- IEC 61131 Programovatelné řídicí jednotky;
- IEC 61175 Označování signálů a spojů;
- IEC 61346 Průmyslové systémy, instalace a zařízení a průmyslové produkty – Zásady strukturování a referenční označování [56].

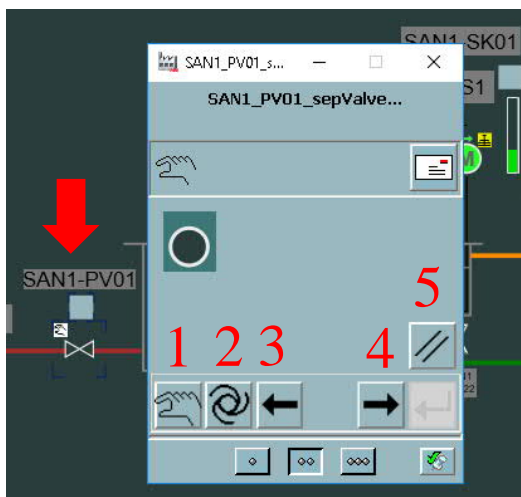
5.3 Grafické rozhraní lapáku písku

Návrh grafického rozhraní pro technologii lapáku písku je vidět na Obr. 50. Pro přepínání mezi jednotlivými grafikami jsou využívány **Aspekt Linky** (označeny červenou šipkou). Aspekt link funguje jako tlačítko, které navede operátora na grafiku, která je s daným aspektem provázána.



Obr. 50: Grafické rozhraní pro technologii lapáku písku

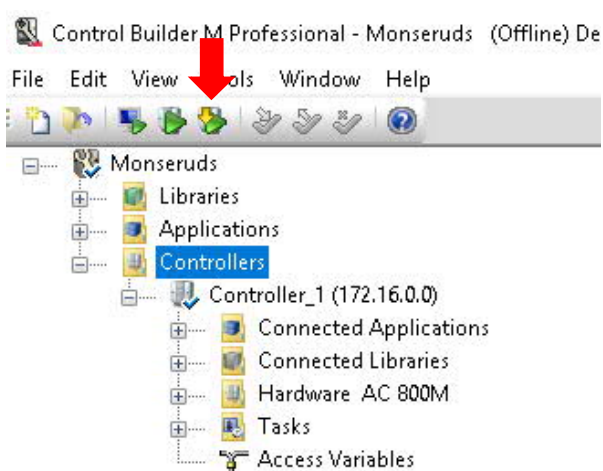
Některé objekty obsahují **Faceplate**, tedy prostředí pro interakci operátora s objektem. Tento Faceplate je otevřen kliknutím na daný objekt. Příklad pro ventil je vidět na Obr. 51. Pod symbolem ruky (označeno číslem 1) je zvoleno manuální ovládání ventilu. Symbol pod číslem 2 vrací ventil do automatického režimu. Výchozí režim ventilu při startu systému je definován z control builderu. Po spuštění tak může být ventil v automatickém režimu nebo v manuálním. V manuálním režimu je možno šipkami 3 a 4 ventil zavírat a otevírat. Symbolem pod číslem 5 schvaluje operátor případné alarmy.



Obr. 51: Faceplate pro ovládání ventilu

6 Testování

Pro testování správnosti naprogramování logik byl použit softwarový kontrolér, který je součástí systému 800xA. Softwarový kontrolér simuluje chování skutečného kontroléru. Podmínkou je tedy spuštěný program Softwarového kontroléru a spuštěný OPC server pro komunikaci s kontrolérem. Testování probíhá v prostředí Control Builderu. Hardware, který je simulován jeho softwarovou náhradou je vybrán uživatelem a označen jako Simulate Hardware (přes pravý klik myši na daný Hw a následně volbou Properties / Simulate Hardware). Control builder obsahuje volbu pro stažení programu do kontroléru a přechod do režimu Online (označeno šipkou na Obr 52).



Obr 52: Stažení programu do kontroléru

V online režimu má možnost uživatel pozorovat hodnoty jednotlivých proměnných v reálném čase. V případě softwarového kontroléru je možno simulovat hodnoty vstupních proměnných (tyto hodnoty jsou v případě skutečného kontroléru získávány ze snímačů a aktuátorů) a následně pozorovat odezvy na výstupních proměnných. Testování na softwarovém kontroléru umožňuje připravit program do jisté z části otestované podoby, nicméně pro dokončení projektu je nutné otestovat program na skutečném technologickém celku a některé parametry doladit až na místě, kde programátor tráví podle rozsahu projektu zpravidla více týdnů až měsíců. Proces uvádění programu do provozu na skutečném celku je známý pod pojmem commissioning. Pro testování na softwarovém kontroléru byly vytvořeny tabulky (Tabulka 4, Tabulka 5), do kterých je zapisováno, zda program vyhovuje jednotlivým podmínkám či nikoliv, popřípadě přidán komentář.

Tabulka 4: Testování lapáku písku

Předmět testování	Očekávaný výsledek	Výsledek	Komentář
Simulace stavu hladiny FET1_LT01 na 0,3 m.	Je spuštěn povel pro otevření poklopu?	Ano	Pouze pokud je poklop zavřen (GS2=True)
Simulace signálu otevřené polohy poklopu ML01.	Přijde po jedné minutě povel pro zavření poklopu?	Ano	
	Generuje chyba na pohonu poklopu ML01 alarm?	Ano	
Simulace domácí polohy škrabáku. Nahromaděné množství průtoku UTL01_FT01 simulováno na 201 m3.	Je aktivován povel pro start škrabáku (SAN1_SK01)?	Ano	
	Je aktivován povel pro start dmychadel (BM01, BM02)?	Ano	
	Je resetováno nahromaděné množství průtoku UTL01_FT01?	Ano	V okamžiku jeho dosažení.
Simulace stavu, kdy škrabák opustil výchozí (GS01 rozepnuto).	Je požadavek na SAN1_SK01 = 40 % otáček pro čas T1 (10 s) a po uplynutí 80 %?	Ano	
	Simulace stavu, kdy projde škrabák domácí polohou GS01. Rychlost SAN1_SK01 opět nastavena na 40 %?	Ano	
	Definováno jako 1 cyklus pro čítač cyklů?	Ano	
Simulace stavu, kdy proběhlo 5 cyklů škrabáku.	Je škrabák zastaven v domácí pozici?	Ano	
	Dmychadla jsou vypnuta?	Ano	
Simulace stavu překročení proudových hodnot ochran proti přetížení WT1.	Při 40 % pohonu generuje alarm hodnota proudu 9,6 mA high alarm a 13,92 mA high high alarm?	Ano	
	Při 80 % pohonu generuje alarm hodnota proudu 10,08 mA high alarm a 14,4 mA high high alarm?	Ano	
Simulace stavu, kdy škrabák SAN1_SK01 nebyl spuštěn za poslední hodinu.	Je aktivován povel pro spuštění SAN1_SK01?	Ano	
	Je resetováno měření času od posledního spuštění?	Ano	
Simulace stavu, kdy škrabák SAN1_SK01 dosáhl 10 cyklů.	Je aktivován povel pro otevření ventilu SAN1_SK01_PV01?	Ano	

Simulace stavu, kdy je ventil SAN1_SK01_PV01 otevřen a škrabák dosáhl 10 cyklů.	Je aktivní povel pro start pumpy SAN1_SK01_PU01?	Ano	
	Je aktivní povel pro zastavení pumpy SAN1_SK01_PU01 po 20 vteřinách od spuštění?	Ano	
Simulace stavu, kdy čerpá pampa SAN1_SK01_PU01, a ventil je náhle uzavřen.	Čerpání je zastaveno, je-li ventil SAN1_SK01_PV01 uzavřen?	Ano	
Simulace stavu, kdy byla vypnuta pampa SAN1_SK01_PU01.	Je aktivní požadavek na uzavření ventilu SAN1_SK01_PV01 po 10 vteřinách od vypnutí pumpy?	Ano	
Simulace stavu, kdy se škrabka nevrátila do domácí polohy.	Je indikován alarm?	Ano	

Tabulka 5: Testování lapáku tuku

Předmět testování	Očekávaný výsledek	Výsledek	Komentář
Simulace stavu, kdy je škrabka na tuk ve výchozí pozici (GS01) a bylo dosaženo nahromaděného množství 200 m ³ na průtokoměru UL0-FT01.	Je spuštěna škrabka na tuk (75 % pohonu)?	Ano	
	Je vyresetováno akumulované množství průtoku na měření UTLO-FT01?	Ano	
	Má škrabka krátkou přestávku (2 s) před změnou směru v koncové poloze GS2?	Ano	
Simulace jednoho dokončeného cyklu a dosažení výchozí polohy GS1.	Je škrabák po 1 cyklu zastaven?	Ano	
Simulace tří dosažených cyklů škrabáku a dosažení výchozí polohy GS1.	Je spuštěno proplachování (SV01) po dobu 5 s?	Ano	Jen pokud je aktivní feedback pohonu FET1_SK01 vypnuto.
Simulace stavu, kdy přijde požadavek na proplachování SV01 v době kdy je spuštěn škrabák tuku FET1.	Ventil SV01 musí zůstat uzavřený.	Ano	Proplachování spuštěno ve chvíli, kdy doběhne pohon škrabáku FET1_SK01.
Simulace stavu, kdy hladina v nádrži LT03 přesáhla 2,5 metru.	Došlo k otevření ventilů FET1-PV02 a FET0-PV02?	Ano	

Simulace stavu, kdy jsou ventily FET0-PV02 a FET1-PV02 otevřeny a hladina PV02 je větší než 2,5m.	Je spuštěna pumpa FET1-PU01 (běh na 75%)?	Ano	
Měření QT01 indikuje změnu hustoty kapaliny. Hustota stoupla nad 30 %.	Je zastaveno čerpání PU01?	Ano	
Hladina FET1-LT03 dosáhla dolní bezpečnostní úrovně pro zastavení čerpadla (1 m).	Je zastaveno čerpadlo FET1-PU01?	Ano	
Tukové čerpadlo FET1-PU01 přešlo ze zapnutého do vypnutého stavu.	Je zavřen ventil FET1-PV02?	Ano	Po uplynutí nastavené prodlevy (2s).
Tukové čerpadlo FET1-PU01 přešlo ze zapnutého do vypnutého stavu.	Je zavřen ventil FET0-PV02?	Ano	Po uplynutí nastavené prodlevy (2s).

Tímto způsobem bylo otestováno pouze zda program splňuje vlastnosti a požadavky očekávané zákazníkem. Výhodou je fakt, že použité funkční bloky jsou již otestovány a validovány. V případě vytváření a užívání vlastních bloků je testování složitější, protože si programátor musí být jist, že vytvořené bloky fungují. Obecně je dobré rozdělit testy na několik úrovní [59] :

- Jednotlivé zkoušky – konkrétní část kódu je zkoušena samostatně.
- Integrovaní testování – jednotlivé části jsou již sestaveny a zkoušeny jsou globálně.
- Testování systému – testována je celá aplikace systému [59].

Podle standardu testování průmyslové automatizace ISA-62381 se provádí testování ve třech fázích [60]:

- Výrobní přejímací zkouška (FAT);
- Přejímací zkouška na místě (SAT);
- Zkouška začlenění v místě (SIT).

Dále je možno testy rozdělit na automatizované a ruční. V některých případech není dost dobře možné ručně otestovat možné situace a přichází tak na řadu testování automatizované. Manuální a automatizované testy se vzájemně doplňují. Tyto testy jsou doplňkem a nenahrazují výše uvedené metody [59].

7 Závěr

Náplní práce bylo objasnit princip návrhu řídicího systému pro konkrétní část technologického celku. Aby mohl být tento cíl splněn musel být technologický proces charakterizován do formy funkčních popisů, které jasně definují funkci jednotlivých částí. Tím byl splněn požadavek prvního bodu zadání. Na celek čistírny se svým rozsahem (počet zařízení, počet I/O signálů...) bylo vhodné využít distribuovaný systém řízení. Distribuovaný systém byl zvolen proto, aby nemusela jediná jednotka CPU provádět veškeré výpočty. Definování požadavků obsažené v práci je také odpovědí na první bod zadání této práce.

Druhým bodem zadání bylo navrhnout konfiguraci řídicího systému. Systém pro komunikaci mezi PLC a frekvenčním měničem využívá komunikačního standardu PROFIBUS DP. Profibus kabel pro přenos informací je složen pouze ze dvou vodičů a stínění, díky tomu bylo ušetřeno množství drátových spojení. Zvoleným PLC kontrolérem je kontrolér AC 800M společnosti ABB. Alokace signálů byla provedena tak, aby se minimalizovali možnosti selhání systému díky využití principů redundance. Pro řízení celku byl vybrán systém 800xA. Tento systém je vhodný pro využití na celek takového rozsahu jako byl tento projekt a také programové knihovny systému nabízí potřebné bloky. Dále umí komunikovat za pomoci PROFIBUS protokolu (ale i dalších) a splňuje požadavky na redundanci. Splňuje také požadavek na více operátorských pracovišť a z tohoto důvodu se stal vhodným výběrem. Kapitola 3.8 je věnovaná konfiguraci systému a je odpovědí na druhý bod zadání.

Cíle práce uvedené ve třetím bodě zadání byli splněny realizací funkčních popisů, popisem systému značení, vytvořením tabulky vstupů a výstupů a vytvořením vývojových algoritmů. Ke vstupům a výstupům byli přiřazeny konkrétní kanály na vstupních a výstupních modulech.

Navržené algoritmy byli dle požadavku čtvrtého bodu zadání implementovány do systému. Postup implementace algoritmů je popsán v kapitole 4. Postup tvorby grafického rozhraní je popsán v kapitole 5. Návrh vizualizace byl postaven tak, aby operátor přepínal mezi obrazovkami pro technologii zachytávání písku a zachytávání tuku. Na jedné obrazovce se tak neobjevuje přebytek informací a pozice jednotlivých zařízení se snaží napodobit sestavu reálné technologie.

Testování požadované pátým bodem zadání je obsaženo v kapitole 6. Výhodou zvoleného systému byla možnost testování za pomoci softwarového kontroléru. Toto umožnilo simulovat

hodnoty I/O proměnných softwarově. Bez softwarového kontroléru by bylo nutné simulovat hodnoty signálů za pomoci skutečných fyzikálních veličin. Testování propojení fyzických signálů – takzvaný Loop testing probíhá až na místě stavby, toto nebylo možné otestovat. Během testování se objevily problémy s komunikací mezi systémem a softwarovým kontrolérem a musel být restartován OPC server, který tuto komunikaci obsluhuje. Některé v programu přednastavené konstanty (například časové konstanty) by v průběhu uvádění do provozu byly pravděpodobně upraveny podle potřeb skutečného celku, popřípadě by bylo umožněno operátorovi některé konstanty měnit (například počet provozních cyklů technologie po jejím spuštění). Program je připraven do podoby, která splňuje očekávání funkčního popisu.

Seznam použité literatury

- [1] LACKO, Branislav. *Automatizace a automatizační technika 1 : systémové pojetí automatizace*. Brno: Computer Press, 2012. ISBN 978-80-251-3628-7.
- [2] BRUSSO, Barry C. 50 Years of Industrial Automation [History]. *IEEE Industry Applications Magazine* [online]. 2018, **24**(4), 8–11. ISSN 1077-2618. Dostupné z: doi:10.1109/MIAS.2018.2820440
- [3] GOLDFARB, Glenn. Hard to find maintenance tips for electromechanical relays. In: *2017 70th Annual Conference for Protective Relay Engineers (CPRE)* [online]. B.m.: IEEE, 2017, s. 1–6. ISBN 978-1-5386-1581-2. Dostupné z: doi:10.1109/CPRE.2017.8090026
- [4] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace. 1. díl, Základní pojmy, úvod do programování*. 1. Praha: BEN - technická literatura, 1999. ISBN 978-80-86056-58-6.
- [5] ZEZULKA, František. *Prostředky průmyslové automatizace. 1*. Brno: Vysoké technické učení v Brně, 2004.
- [6] SARABATY, H. *Internal organization of computers : Memories & CPU Architecture*. 2017.
- [7] ABB. *ABB Ability™ System 800xA Control and I/O Overview*. 3BSE047351 en K. 2011.
- [8] BINDAL, Ahmet. System Bus. In: *Fundamentals of Computer Architecture and Design* [online]. Cham: Springer International Publishing, 2019, s. 119–150. Dostupné z: doi:10.1007/978-3-030-00223-7_4
- [9] BENEŠ, Pavel. *Automatizace a automatizační technika. 3, Prostředky automatizační techniky*. Praha: Computer Press, 2014. ISBN 80-7226-248-3.
- [10] *Automation - Manufacturing applications of automation and robotics / Britannica.com* [online]. [vid. 2018-10-10]. Dostupné z: <https://www.britannica.com/technology/automation/Manufacturing-applications-of-automation-and-robotics>
- [11] CHENGUANG GUO, YANLONG ZHANG, LEI CHEN, TAO ZHOU, XUEWU LI, MIN WANG a ZHIPING WEN. A novel application of FPGA-based partial dynamic reconfiguration system with CBSC. In: *2012 VIII Southern Conference on Programmable Logic* [online]. B.m.: IEEE, 2012, s. 1–4. ISBN 978-1-4673-0186-2. Dostupné z: doi:10.1109/SPL.2012.6211794

- [12] *TYPES OF AUTOMATION in Production and Operations Management (9679) / Wisdom Jobs* [online]. [vid. 2018-10-10]. Dostupné z: <https://www.wisdomjobs.com/e-university/production-and-operations-management-tutorial-295/types-of-automation-9679.html>
- [13] BACKLUND, Alexander. *The definition of system* [online]. London: Computers--Cybernetics, 2000. ISBN 0368492X. Dostupné z: <https://search.proquest.com/docview/213909631?accountid=14965%0A>
- [14] *What is Industrial Automation / Types of Industrial Automation* [online]. [vid. 2018-10-06]. Dostupné z: <https://www.electricaltechnology.org/2015/09/what-is-industrial-automation.html>
- [15] *Field, Control, Supervisory, and Enterprise Levels of Automation* [online]. [vid. 2018-10-07]. Dostupné z: <https://www.machinedesign.com/industrial-automation/differences-between-field-control-supervisory-and-enterprise-levels-automation>
- [16] *Choosing between centralized and distributed control system designs / Control Engineering* [online]. [vid. 2018-10-12]. Dostupné z: <https://www.controleng.com/single-article/choosing-between-centralized-and-distributed-control-system-designs/60f079d8d1a38dfae835d05d5e643768.html?OCVALIDATE=>
- [17] *Centralized vs. distributed control / Control Engineering* [online]. [vid. 2018-10-12]. Dostupné z: <https://www.controleng.com/single-article/centralized-vs-distributed-control.html>
- [18] *Centralized vs Distributed Version Control Systems - What's Best for Notes Development? — Teamstudio* [online]. [vid. 2019-05-28]. Dostupné z: <https://www.teamstudio.com/blog/distributed-vs-centralized-version-control-systems-for-lotus-notes>
- [19] *Four reasons to consider distributed vs. centralized control / Eaton* [online]. [vid. 2019-05-28]. Dostupné z: <https://www.eaton.com/us/en-us/services/dynamic-machine-control/dmc-articles/4-reasons-to-consider-distributed-vs-centralized-control.html>
- [20] BOHM, Pavel. *Komunikační modul pro průmyslovou sběrnici Foundation Fieldbus*. B.m., 2012. Západočeská univerzita v Plzni.
- [21] Æ, Peter Neumann. Communication in industrial automation — What is going on ? [online]. 2007, **15**, 1332–1347. Dostupné z: doi:10.1016/j.conengprac.2006.10.004

- [22] *Průmyslová komunikace* [online]. [vid. 2019-02-17]. Dostupné z: <https://www.foxon.cz/produkty/prumyslova-komunikace>
- [23] GABOR, Georgel, Cosmin PINTILIE, Catalin DUMITRESCU, Nituca COSTICA a Adrian Traian PLESCA. Application of Industrial PROFIBUS-DP Protocol. *2018 International Conference and Exposition on Electrical And Power Engineering (EPE)*. 2018, 614–617.
- [24] ALVES, Mário a Eduardo TOVAR. Engineering PROFIBUS networks with heterogeneous transmission media. *Computer Communications* [online]. 2006, **30**(1), 17–32. ISSN 01403664. Dostupné z: doi:10.1016/j.comcom.2006.07.011
- [25] DRAHOŠ, Peter a Igor BÉLAI. The PROFIBUS protocol observation. *IFAC Proceedings Volumes (IFAC-PapersOnline)* [online]. 2012, **9**(PART 1), 258–263. ISSN 14746670. Dostupné z: doi:10.3182/20120619-3-RU-2024.00058
- [26] *Monserud rensanlegg - RINGVIRKNINGER* [online]. [vid. 2018-10-17]. Dostupné z: <http://www.ringvirkninger.com/prosjekter/monserud-rensanlegg/>
- [27] *VŠB TU Ostrava* [online]. [vid. 2018-11-04]. Dostupné z: http://homen.vsb.cz/hgf/546/materialy/radka_2010/mc.html
- [28] DUDÍKOVÁ, Kristýna. *The comparison of two waste water treatment systems - constructed wetlands and mechanical-biological waste water treatment plant*. B.m., 2010. Univerzita Karlova v Praze.
- [29] *Lapák písku - BMTO GROUP a.s.* [online]. [vid. 2018-10-28]. Dostupné z: <https://www.bmto.cz/separace-a-prani-pisku/lapak-pisku/>
- [30] DOŠEK, Michal. *Rozbor funkce čistírny odpadních vod*. B.m., 2008. Mendelova zemědělská a lesnická univerzita v Brně.
- [31] *ISO 15519-2:2015(en), Specifications for diagrams for process industry — Part 2: Measurement and control* [online]. [vid. 2018-11-19]. Dostupné z: <https://www.iso.org/obp/ui/#iso:std:iso:15519:-2:ed-1:v1:en:sec:E>
- [32] *P&IDs (Piping & Instrumentation Diagrams) and P&ID Valve Symbol Library - Assured Automation* [online]. [vid. 2018-11-22]. Dostupné z: <https://assuredautomation.com/news-and-training/pids-piping-instrumentation-diagrams-and-pid-valve-symbol-library/>
- [33] *S800 I/O - Comprehensive range of high performance I/Os for automation solutions (ABB Compact Product Suite for building your own DCS)* [online]. [vid. 2019-03-08]. Dostupné z: <https://new.abb.com/control-systems/essential-automation/compact-product-suite/essential-connect-suite/s800IO>

- [34] *Pluto - Programovatelné bezpečnostní řídicí jednotky (Bezpečnostní systémy) / ABB* [online]. [vid. 2019-03-08]. Dostupné z: <https://new.abb.com/low-voltage/cs/nizke-napeti/produkty/bezpecnostni-systemy/programovatelne-bezpecnostni-ridici-jednotky/pluto>
- [35] *ABB DCS PM866 CPU ABB AC800M PLC AC 800M Controller* [online]. [vid. 2019-03-08]. Dostupné z: <https://www.aotewell.com/abb/abb-dcs-pm866-cpu-abb-ac800m-plc-ac-800m-controller.html>
- [36] *ABB 800xA DCS distributed control system - ABB System 800xA - process, electrical, safety, telecoms in one system* [online]. [vid. 2019-03-20]. Dostupné z: <https://new.abb.com/control-systems/system-800xa/800xa-dcs>
- [37] MEHTA, B.R. a Y.J. REDDY. Distributed control system. In: *Industrial Process Automation Systems* [online]. B.m.: Elsevier, 2015, s. 75–133. Dostupné z: doi:10.1016/B978-0-12-800939-0.00006-1
- [38] *ABB Ability™ System 800xA 6.0.3.2 System Guide Summary* [online]. nedatováno [vid. 2019-03-09]. Dostupné z: https://library.e.abb.com/public/a828bb89535449829cabd8a60fe804ad/3BSE078159enE_ABB_Ability_System_800xA_6.0.3.2_System_Guide_Summary.pdf
- [39] ABB. *AC 800M OPC Server*. 3BSE035983-510. 2010.
- [40] CRNKOVIC, Ivica a Magnus LARSSON. Component-based Software Engineering- New Paradigm of Software Development. 2001, (January).
- [41] *ABB Aspect Objects Technology - ABB 800xA DCS distributed control system (ABB System 800xA - process, electrical, safety, telecoms in one system)* [online]. [vid. 2019-03-11]. Dostupné z: <https://new.abb.com/control-systems/system-800xa/800xa-dcs/system/aspect-object-technology-industrial-it>
- [42] ABB. *System 800xA engineering: Overview*. 3BDD013082. 2010.
- [43] ABB. *KE-90 . 4600 Process Automation project work - ABB 800xA introduction*. T314 Course (Interní dokumentace ABB). 2010.
- [44] *Control Builder - PLCs (Automation Builder) / ABB* [online]. [vid. 2018-12-01]. Dostupné z: <https://new.abb.com/plc/automationbuilder/plcs/control-builder>
- [45] ABB. *Getting Started, System 800xA Control* [online]. 3BSE041880-600 A. 2016. Dostupné z: <http://search.abb.com/library/download.aspx?documentid=3bse041880-510&languagecode=en&documentpartid=&action=launch>
- [46] ABB. *Utility Library User 's Manual*. 1KGF100649-510. 2010.
- [47] ČSN. *Česká technická norma ČSN EN 61131-3 ed. 2*. 2013.

- [48] ABB. *Function Description for Signal Object sesDigInM*. 1KGF100451-512. 2011.
- [49] ABB. *System 800xA Control AC 800M Planning* [online]. 3BSE043732-510. 2013.
Dostupné z: <http://search.abb.com/library/download.aspx?documentid=3bse041880-510&languagecode=en&documentpartid=&action=launch>
- [50] ABB. *Function Description for Process Object sepMotVarM Motor Variable Speed*. 1KGF100686-512. 2011.
- [51] ABB. *AC 800M PROFIBUS DP Configuration*. 3BDS009030-510 A. 2013.
- [52] ABB. *User's Manual PROFIBUS DP Adapter Module RPBA-01*. 2004, 0–1.
- [53] ACROMAG. *Introduction to Profibus DP*. 8500–698–A02M000. 2002.
- [54] ABB. *AC 800M PROFIBUS DP Installation*. 3BDS009029R5001 B. 2011.
- [55] ABB. *System 800xA Engineering Process Graphics*. 3BSE049230-510 E. 2010.
- [56] ABB. *Utility Library for AC 800M Function Description for Alarm handling*. 1KGF100653-512. 2011.
- [57] ČSN EN 60073 - *Náhled* [online]. [vid. 2019-04-18]. Dostupné z: http://csnonlinefirmy.unmz.cz/html_nahledy/33/56073/56073_nahled.htm
- [58] HOSSAIN, Akram a Tanima ZAMAN. AC 2012-3605 : HMI DESIGN : AN ANALYSIS OF A GOOD DISPLAY FOR SEAMLESS INTEGRATION BETWEEN USER UNDERSTANDING AND HMI Design : An Analysis of a Good Display for Seamless Integration Between User Understanding and Automatic Controls Abstract : In process a. *American Society for Engineering Education*. 2012, 14.
- [59] FERNÁNDEZ, B a E BLANCO. Testing & verification of PLC code for process control. *Proc. of the 14th Int. Conf.* [online]. 2013, (October 2013). Dostupné z: <http://accelconf.web.cern.ch/AccelConf/ICALPCS2013/papers/thppc080.pdf>
- [60] ČSN EN 62381 (180412) - *Technické normy ČSN* [online]. [vid. 2019-04-25]. Dostupné z: http://www.technicke-normy-csn.cz/180412-csn-en-62381_4_79892.html

Seznam obrázků

OBR. 1: STRUKTURA PLC (PŘEKRESLENO Z [9])	13
OBR. 2: VÝBĚR VHODNÉHO SYSTÉMU (PŘEKRESLENO Z [12])	15
OBR. 3: BLOKOVÝ SCHÉMA ŘÍZENÍ S PLC (PŘEKRESLENO Z [9])	15
OBR. 4: HIERARCHICKÉ USPOŘÁDÁNÍ SYSTÉMU PRŮMYSLOVÉ AUTOMATIZACE (PŘELOŽENO Z [14]).....	16
OBR. 5: PROCES TVORBY SYSTÉMU	20
OBR. 6: VÝVOJOVÝ DIAGRAM PRO LAPÁK PÍSKU	28
OBR. 7: VÝVOJOVÝ DIAGRAM OCHRANY PŘED PŘETÍŽENÍM ŠKRABÁKU	29
OBR. 8: VÝVOJOVÝ DIAGRAM FUNKCE ČERPÁNÍ PUMP	30
OBR. 9: VÝVOJOVÝ DIAGRAM FUNKCE POKLOPU	31
OBR.10: PŘÍKLAD ZÁLOHOVANÉHO ZAŘÍZENÍ.....	32

OBRÁZEK 11: KONCEPT OBJEKTU A ASPEKTU (PŘEKRESLENO Z [41])	33
OBR. 12: NÁHLED DO STRUKTURY SYSTÉMU 800xA	34
OBR. 13: PLANT EXPLORER	34
OBR. 14: SNÍMEK OBRAZOVKY CONTROL BUILDERU	36
OBR. 15: PŘIDÁNÍ KNIHOVNY DO PROJEKTU	37
OBR. 16: PROSTŘEDÍ PRO NASTAVENÍ KONTROLÉRU	38
OBR. 17: IDENTITA KONTROLÉRU	38
OBR. 18: VKLÁDÁNÍ I/O MODULŮ	39
OBR. 19: PŘIŘAZENÍ TASK CONNECTION	40
OBR. 20: FUNKČNÍ DIAGRAM	41
OBR. 21: ZÁPIS PROMĚNNÉ	42
OBR. 22: ALOKACE SIGNÁLU NA KARTU	42
OBR. 23: ALOKOVANÉ SIGNÁLY	43
OBR. 24: ZMĚNA ROZSAHU SIGNÁLU	43
OBR. 25: NASTAVENÍ TYPU SIGNÁLU	44
OBR. 26: PŘIDÁVÁNÍ OBJEKTŮ DO FUNKČNÍHO DIAGRAMU	45
OBR. 27: BLOK VLOŽENÝ DO POLE FUNKČNÍHO DIAGRAMU	45
OBR. 28: PARAMETRY BLOKU SESDIGINM	46
OBR. 29: BLOČEK OBSLUHUJÍCÍ DIGITÁLNÍ SIGNÁL	46
OBR. 30: BLOČEK ČÍTAČE	47
OBR. 31: MĚŘENÍ PRŮTOKU	47
OBR. 32: NASTAVENÍ ČASOVAČE	48
OBR. 33: POUŽITÍ FUNKCE JOIN	49
OBR. 34: VYTVOŘENÍ NOVÉ KNIHOVNY	49
OBR. 35: DEVICE IMPORT WIZARD	50
OBR. 36: VOLBA TYPU TELEGRAMU	51
OBR. 37: KONFIGURACE PROFIBUSU	52
OBR. 38: KONFIGURACE PROFIBUS KOMUNIKACE	52
OBR. 39: ADAPTÉR RPBA-01	53
OBR. 40: NĚKTERÉ PARAMETRY STATUS WORDU	53
OBR. 41: ALOKACE SIGNÁLŮ NA PROFIBUS	54
OBR. 42: PROFIBUS SIGNÁLY NAVÁZANÉ NA BLOČEK MOTORU	55
OBR. 43: OVLÁDÁNÍ MOTOROVÉHO BLOKU Z LOGIKY	55
OBR. 44: OBLASTI PRO VYTVÁŘENÍ GRAFIK	57
OBR. 45: ASPEKT GRAFICKÉHO DISPLEJE	57
OBR. 46: GRAFICKÝ OBJEKT VLOŽENÝ DO PROSTŘEDÍ GRAPHICS BUILDERU	58
OBR. 47: RŮZNÉ STAVY MOTORU	59
OBR. 48: PROGRAMOVÁNÍ PODMÍNEK V GRAFICKÉM OBJEKTU	59
OBR. 49: INDIKOVÁN HIGH HIGH ALARM (PŘETÍŽENÍ NA OCHRANĚ WT1)	60
OBR. 50: GRAFICKÉ ROZHRAŇÍ PRO TECHNOLOGII LAPÁKU PÍSKU	61
OBR. 51: FACEPLATE PRO OVLÁDÁNÍ VENTILU	62
OBR. 52: STAŽENÍ PROGRAMU DO KONTROLÉRU	63

Seznam tabulek

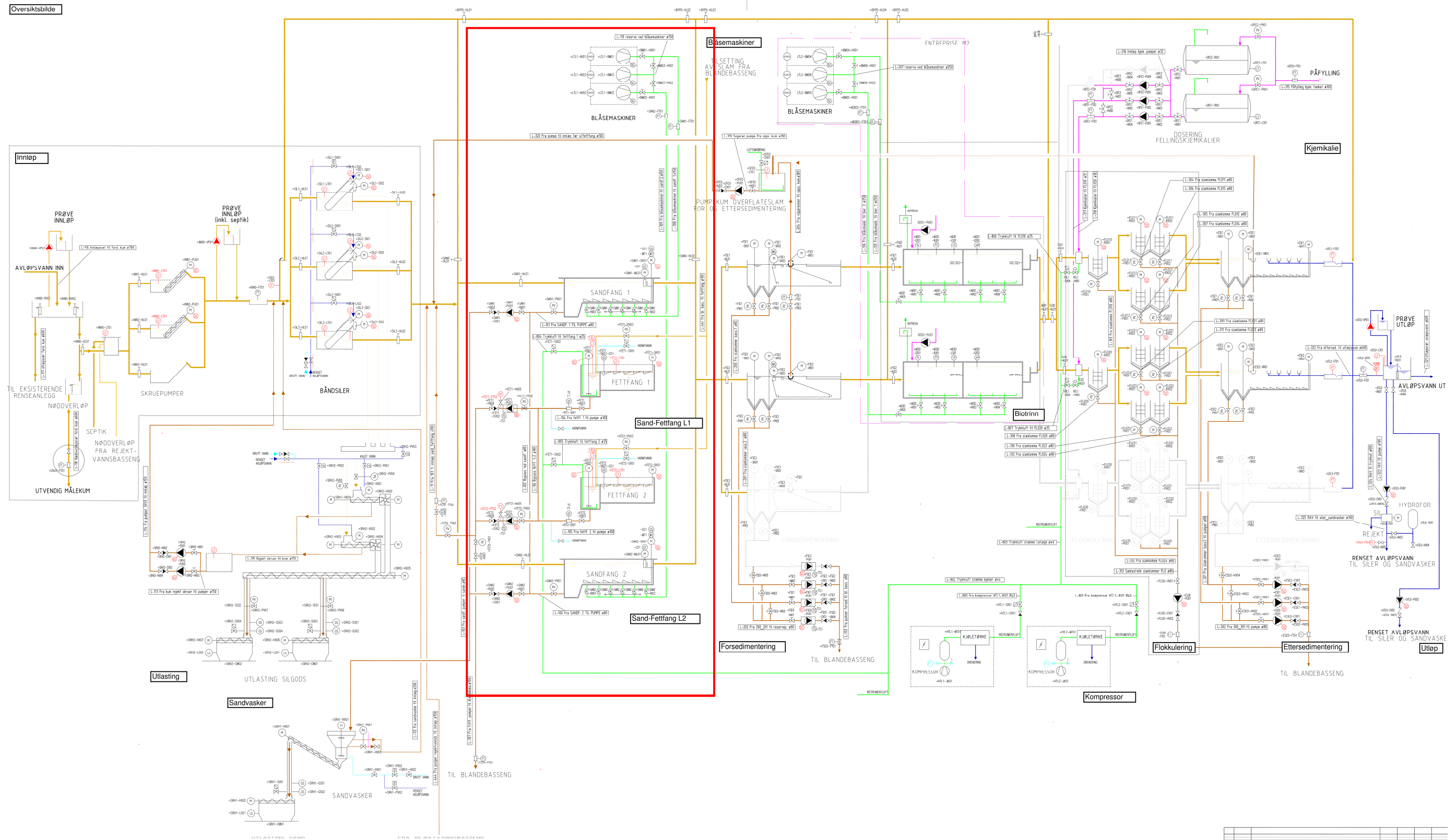
TABULKA 1: SEZNAM OBJEKTŮ [ZDROJ: FUNKČNÍ POPIS PROJEKTU - PŘIDÁN DO PŘÍLOHY]	24
TABULKA 2: TYPY SIGNÁLŮ [ZDROJ: FUNKČNÍ POPIS PROJEKTU - PŘIDÁN DO PŘÍLOHY]	25
TABULKA 3: MOŽNOSTI SYSTÉMU 800xA	35
TABULKA 4: TESTOVÁNÍ LAPÁKU PÍSKU	64
TABULKA 5: TESTOVÁNÍ LAPÁKU TUKŮ	65

Seznam příloh

PŘÍLOHA 1: PIPING AND INSTRUMENTATION DIAGRAM (P&ID).....	76
PŘÍLOHA 2: HIERARCHICKÉ USPOŘÁDÁNÍ PROJEKTU	77

Přílohy

Příloha 1: Piping and instrumentation diagram (P&ID)



Příloha 2: Hierarchické uspořádání projektu

