

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra řídicí techniky



Bakalářská práce

Vizualizace technologického procesu sušárny vápence

Visualization of the technological process of the limestone dryer

Autor: Tomáš Vayhel

Vedoucí práce: Ing. Miloš Fořt

Vedoucí práce – konzultant: Ing. Richard Šusta, Ph.D.

Studijní program: Kybernetika a robotika

Praha 2022

20. KVĚTNA 2022

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vayhel** Jméno: **Tomáš** Osobní číslo: **491860**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra řídicí techniky**  
Studijní program: **Kybernetika a robotika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Vizualizace technologického procesu sušárny vápence**

Název bakalářské práce anglicky:

**Visualization of the technological process of the limestone dryer**

Pokyny pro vypracování:

1. Prostudujte proces sušení vápence a jeho stávající vizualizace
2. Navrhněte obrazovky nutné k jeho řízení
3. Realizujte zadání s využitím Siemens SIMATIC TIA WinCC Unified
4. Ověřte funkčnost zadání simulací

Seznam doporučené literatury:

- [1] SIMATIC STEP 7 Basic/Professional V17 and SIMATIC WinCC V17 [online]. 2021 [cit. 2021-9-29]. 2. Dostupné z: <https://support.industry.siemens.com/cs/document/109798671/simatic-step-7-basic-professional-v17-and-simatic-wincc-v17?dti=0&lc=en-U>
- [2] SIMATIC HMI devices Basic Panels [online]. Siemens, 2012 [cit. 2021-9-29]. 04/2012. Dostupné z: <https://support.industry.siemens.com/cs/document/31032678/simatic-hmi-hmi-devices-basic-panels?dti=0&lc=en-US>
- [3] SIMATIC HMI WinCC (TIA Portal) WinCC Engineering V17 – Communication [online]. Siemens, 2021 [cit. 2021-9-29]. 05/2021. Dostupné z: [https://support.industry.siemens.com/cs/document/109794203/simatic-hmi-wincc-\(tia-portal\)-wincc-engineering-v17-%E2%80%93-3-communication?dti=0&lc=en-U](https://support.industry.siemens.com/cs/document/109794203/simatic-hmi-wincc-(tia-portal)-wincc-engineering-v17-%E2%80%93-3-communication?dti=0&lc=en-U)
- [4] SIMATIC HMI WinCC Unified WinCC Engineering V17 – WinCC Unified [online]. Siemens, 2021 [cit. 2021-9-29]. Dostupné z: <https://support.industry.siemens.com/cs/document/109794204/simatic-hmi-wincc-unified-wincc-engineering-v17-%E2%80%93-3-wincc-unified-?dti=0&lc=en-U>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Miloš Fořt SIDAT, spol. s r. o., Jinonická 80, 158 00 Praha 5**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

**Ing. Richard Šusta, Ph.D. katedra řídicí techniky FEL**

Datum zadání bakalářské práce: **05.10.2021**

Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce:

**do konce letního semestru 2022/2023**

Ing. Miloš Fořt  
podpis vedoucí(ho) práce

prof. Ing. Michael Šebek, DrSc.  
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.  
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## Čestné prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze dne .....

.....

Podpis autora práce

## Poděkování

Tímto bych chtěl poděkovat Ing. Richardu Šustovi, Ph.D., který se ujal pedagogického dohledu této externí bakalářské práce a byl otevřený konzultacím a dotazům. Dále bych chtěl poděkovat Ing. Martinu Cicvarkovi, který mi byl vždy ochotný vysvětlit a odpovídat na dotazy ohledně firemního programového standardu a vývojového prostředí TIA Portálu.

## Abstrakt

Práce se zabývá tvorbou vizualizace technologického procesu sušárny vápence ve vývojovém prostředí SIMATIC WinCC Unified integrovaném v prostředí TIA Portal.

V průběhu práce je teoreticky popsán výrobní proces a uvedeny základní informace k prostředí TIA Portal a WinCC Unified.

Na teoretický popis navazuje praktická část, ve které je popsána struktura a návrh řízení linky, následně je stručně zmíněná struktura řídicího programu a kapitola je zakončena popisem implementace vizualizačních obrazovek.

Celý projekt je následně otestován s využitím simulátoru, kde je pozornost soustředěna hlavně na správnou funkci řízení technologického procesu.

Klíčová slova: TIA Portal, WinCC Unified, vizualizace, PLC, HMI, sušárna vápence

## Abstract

The work deals with the creation of visualization of the technological process of the limestone dryer in the development environment SIMATIC WinCC Unified integrated into the TIA Portal environment.

The production process is theoretically described during the work with basic information about the TIA Portal and WinCC Unified environment.

The theoretical description is followed by a practical part, which describes the structure and design of line control. The structure of the control program is briefly mentioned, and the chapter ends with a description of the implementation of visualization screens.

The whole project is then tested using a simulator, where attention is focused mainly on the proper functioning of the technological process control.

Keywords: TIA Portal, WinCC Unified, visualization, PLC, HMI, limestone dryer

## Obsah

Úvod .....	7
1. Teoretická část.....	8
1.1. Popis výrobního procesu .....	8
1.1.1 Využití sušení vápence .....	8
1.1.2 Rotační sušárna vápence.....	9
1.1.3 Princip funkce filtru.....	11
1.2. TIA Portál.....	13
1.2.1 PLC.....	13
1.2.2 PLC Tagy .....	13
1.2.3 Organizační bloky v PLC .....	14
1.2.4 Datové bloky .....	15
1.2.5 Funkce .....	16
1.2.6 Funkční bloky.....	16
1.2.7 Formát ukládání dat do paměti.....	16
1.2.8 Instrukce Cont_C.....	17
1.3. Vizualizační prostředí WinCC .....	18
1.3.1 Popis HMI .....	18
1.3.2 SIMATIC HMI Unified Control Panel.....	19
1.3.3 HMI tagy .....	19
1.3.4 Textové listy .....	20
1.3.5 Grafické listy .....	21
1.4. Komunikace.....	22
1.4.1 ProfiBus.....	22
1.4.2 ProfiNet .....	22
2. Praktická část.....	24
2.1. Struktura výrobní linky .....	24
2.1.1 Analýza technologického schéma .....	24
2.1.2 Popis řízení filtru .....	25
2.1.3 Popis řízení rotační sušárny.....	26
2.2. Návrh řídicího programu .....	26
2.2.1 Základní konfigurace projektu v TIA Portálu .....	27
2.2.2 Struktura využitého firemního programového standardu.....	28
2.2.3 Implementace standardizovaných objektů do projektu .....	30
2.2.4 Implementace programových vazeb.....	30



VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

2.2.5	Vyhodnocení funkce dvojitě výklopní klapky filtru .....	34
2.2.6	Regulační obvody.....	34
2.3.	Implementace vizualizace .....	35
2.3.1	Propojení HMI panelu s PLC .....	35
2.3.2	Přístup k datům z PLC .....	36
2.3.3	Návrh struktury vizualizačních objektů .....	36
2.3.4	Tagové listy pro vizualizaci .....	41
2.3.5	Tvorba textových listů.....	42
2.3.6	Statické vizualizační ikony .....	43
2.3.7	Dynamické vizualizační ikony .....	44
2.3.8	Uživatelské rozhraní obrazovek .....	45
2.3.9	Kompletace vizualizačních obrazovek.....	46
3.	Simulace výsledků .....	51
3.1.	Zprovoznění simulátoru obrazovek.....	51
3.2.	Kontrola provázaných signálů pro vizualizaci .....	53
3.3.	Příprava ověření programové funkce .....	53
3.4.	Testování implementované technologické funkce .....	54
4.	Závěr .....	64
	Dodatky .....	65
	Seznam literatury.....	66

# Úvod

Bakalářská práce se bude zabývat řešením praktického projektu realizovaného pod firmou Sidat pro fabriku zákazníka v Rusku. Konkrétně budou dle zadání analyzovány teoretické podklady, popsána tvorba vizualizace technologického procesu sušárny vápence s využitím operátorských panelů Unified a platformy SIMATIC WinCC Unified integrované v prostředí TIA Portal a jejich následné ověření simulací. Nad rámec zadání bude rozebrán i návrh a realizace samotného řídicího programu.

Následující pasáže se proto zaměří na teoretický rozbor jak dané problematiky, tak i využitých principů a nástrojů. Po sekci věnující se teorii následuje praktická část rozebírající možnosti postupu, jejich provedení a zdůvodnění konečně zvoleného řešení. Po praktické části dokument končí ověřením funkce jak výsledného programu, tak vizualizace a kontrolou dodržení všech předepsaných požadavků funkce linky.

Teoretický rozbor je zahájen popisem výrobního procesu jako takového. Bude tedy zmíněno, jaké výhody má proces sušení vápence, následně bude popsána funkce rotační sušárny a princip použité filtrační jednotky. Na popis výrobního procesu navazují informace o využití programovatelné řídicí jednotce a rozbor potřebných informací ohledně vývojového prostředí TIA Portal. Dále budou objasněny a vysvětleny informace ohledně vizualizačního prostředí WinCC Unified a opět popsány využitě nástroje, načež bude teoretický rozbor uzavřen sekcí ohledně komunikace mezi PLC programem a operátorským panelem s nahranou vizualizací.

Po teoretické části následuje sekce zabývající se praktickou realizací, a tedy i odvedenou prací na projektu. Její struktura opět kopíruje podobný formát jako předchozí kapitola, tedy že se nejdříve rozvede analýza projektových podkladů doplněná o konkrétní návrh řízení jak filtru, tak i rotační sušárny spolu s jejich vzájemnou vazbou. Na popis struktury konkrétní výrobní linky navazuje návrh řídicího programu, ve kterém bude spíše letmo popsána tvorba programových vazeb a struktura, jakou byly do projektu zaneseny podmínky řízení technologického procesu zmíněných u popisu řízení jednotlivých sekcí procesu. Po implementaci řídicího programu pokračuje práce popisem tvorby vizualizace procesu. Popis je veden od samotného provázání zobrazovaných informací s daty z PLC, přes návrh metod využitých k tvorbě nového firemního vizualizačního standardu na platformu Unified, i když implementace samotného standardu byla posléze provedena někým jiným. Popis tvorby vizualizace zakončuje využití onoho dokončeného standardu v projektu a následná tvorba vizualizačních obrazovek.

Práce je posléze pro účely zprávy zakončena ověřením dosažených výsledků s využitím simulací jak programu, tak vizualizace. V kapitole bude nicméně rozebráno i to, jak dosáhnout samotného zprovoznění simulace a taky jaká byla potřeba pro účely simulace udělat zjednodušení, spolu s jejich důsledky.

# 1. Teoretická část

Teoretická část se zabývá popisem samotného výrobního procesu sušení vápence, základním popisem použitých komponent a vysvětlením hlavních využívaných pojmů. Výčet využívaných zkratk, spolu s jejich významem je uveden v tabulce číslo 1 níže.

Tab. 1: Zkratky a jejich význam

Zkratka	Význam
PLC	Programovatelný logický automat
HMI / OP	Human Machine Interface / Operátorský panel
TIA Portal	IDE pro Siemens SIMANTIC ve verzi 17
WinCC	Vizualizační prostředí integrované v TIA Portalu
RT	Runtime
Tag	Unikátní identifikátor elementu v databázi
Faceplate	Vizualizační knihovni prototyp objektu

## 1.1. Popis výrobního procesu

Výrobní proces, pro který bude v pozdějších fázích dokumentu popsán návrh řízení a vizualizačních obrazovek se sestává z rotačního sušícího bubnu, filtrační jednotky a spalovací komory. Přísun i odsun materiálů je samozřejmě nedílnou součástí procesu, avšak není z hlediska projektu krom pokynů o připravenosti technologie řízen.

Funkce rotační sušárny, stejně jako filtrace, je rozepsána v bodech níže. Důležité je však zmínit, že uvedená schémata a popisy plně neodpovídají skutečné lince, i když byla snaha o maximální podobnost s ohledem na demonstraci principu funkce.

Pro komoru na rozdíl od zbylých dvou částí je k dispozici pouze její řídicí jednotka s propojovacími signály. Prakticky tedy její řízení probíhá pouze signály ON/OFF a modulací výkonu pece k regulaci teploty výstupního plynu, a tedy v následujících sekcích nebude věnována pozornost její funkci.

### 1.1.1 Využití sušení vápence

Jelikož se řízená linka zabývá sušením vápence, je vhodné zmínit k čemu je vůbec takovýto postup užitečný. Rozbor není sice podstatný z hlediska řízení, může se však hodit jak nadstavbová informace o podstatě samotného procesu.

Vápenec je sedimentární hornina, která má mnohá využití jak v zemědělství na zvyšování pH půdy, tak v průmyslu. Mezi jeho aplikace patří například využití jako hnojivo, či stavební materiál. Proces sušení vápence bývá využit k přípravě materiálu k dalšímu zpracování, vyžadují-li navazující výrobní kroky snížení jeho vlhkost. Vysušením lze totiž předejít problémům jako je ucpávání materiálu v dalším vybavení na zpracování vápence a usnadnit jeho aplikaci. [1][2]

Před zpracováním bývá vápenec po vytěžení nejprve rozdrcen a naskladněn. V případě přílišné vlhkosti materiálu se přistoupí k sušicímu procesu, například s využitím rotační sušárny vápence.

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

Předzpracování materiálu totiž usnadňuje další výrobní kroky, zlepšuje jejich produkci a ve výsledku zvyšuje kvalitu finálního produktu. [1][3]

Jedním z příkladů procesů s potřebnou aplikací kroku sušení je peletizace vápence. Při peletizaci je totiž zapotřebí přidání pojiva k přilínání (adhezi) nadrceného vápence, čímž se zvýší jeho vlhkost a obratem tedy vzniká nutnost přidání sušicího kroku. Jedná se o postup využívaný pro zemědělské aplikace, kde vápenec slouží ke zvýšení pH půdy. Existuje totiž několik forem vápna využívaných ke zvýšení kyselosti půdy a mezi příklady patří: [1][2][4]

- Zemědělské hnojivo: v kontextu zemědělství se jedná o nahrubo nadrcený vápenec. Ačkoliv se snadno aplikuje, jeho nevýhodou je právě nerovnoměrná velikost, která zapříčiní různě dlouhé doby vstřebání do půdy. [2][4]
- Mletý vápenec: jde o jemně nadrcený vápenec, mezi jehož značné výhody patří malá velikost jednotlivých částic. Díky menším kusům je totiž půda ve styku s celkovou větší povrchovou plochou účinné složky materiálu. Naopak nevýhodou práškového vápence je vysoká prašnost, obtížnost aplikace a přenositelnosti. [2][4]
- Granulovaný vápenec: jedná se o peletizovaný jemně nadrcený vápenec. Peletizace probíhá přidáním pojiva do prášku k zvýšení přilínání (adhezi) materiálu a využitím dalšího vybavení k promíchání, granulování a posléze sušení, díky zvýšení vlhkosti přidáním pojiva. Komplikovanější výroba však přináší kombinaci výhod obou předchozích bodů a eliminaci jejich nevýhod. Navrch lze ještě přidat do pelet aditiva, jako například živiny, v procesu mísení s pojivem. [2][4]

Ve výrobních procesech se na sušení vápence běžně využívá rotační sušárna, jelikož lze s její pomocí sušit vápenec jak v peletizované formě, tak i v hrubší podobě. Vyjma variability poskytuje rotační buben i spolehlivé, robustní řešení a rovnoměrné vysušení materiálu. [1][3]

### 1.1.2 Rotační sušárna vápence

Rotační sušárny se využívají k předzpracování materiálu, nebo k dosušování výsledného produktu a lze je aplikovat jak na jemné, tak i hrubé materiály. Jedná se o robustní sušárny s odolným designem [3]. Vizualní model rotační sušárny je k nahlédnutí na obrázku číslo 1.

Jednou z výhod rotační sušárny je její velký rotující cylindr, který zajišťuje vysokou operační kapacitu, která ve srovnání s alternativami umožňuje výrazně větší množství zpracovaného materiálu, a to v rozsahu od 1 tuny za hodinu až i přes 200 tun materiálu za hodinu. V případě peletizovaného materiálu je další výhodou rotačního válce dodatečné vyhlazování produktu [3].

Za provozu je v bubnu sušárny materiál v přímém kontaktu s vysoušecím vzduchem, což v kombinaci s využitím zdvihacích lopatek na stěnách bubnu zvyšuje efektivitu přenosu tepla. Lopatky zároveň slouží k omezení shlukování sušeného materiálu a povzbuzují rovnoměrnost kvality výsledného produktu [3].

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

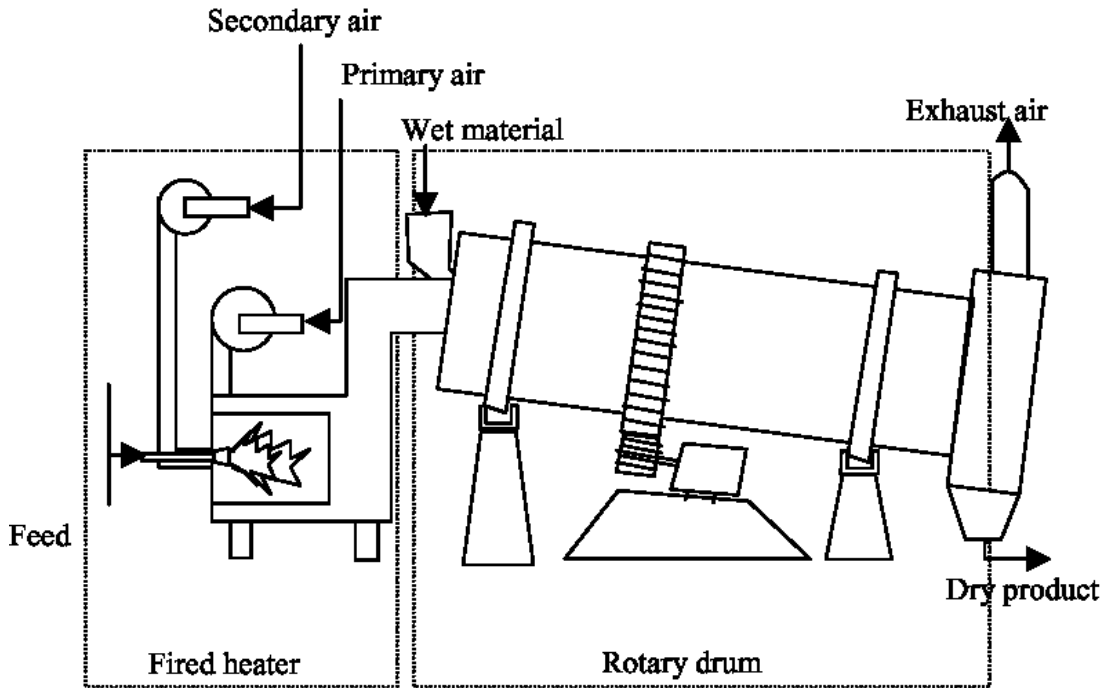


Obr 1: Model rotační sušárny od firmy Feeco pro demonstrativní účely [1]

Na vrh k robustnosti řešení při zpracování jsou tyto sušárny ceněné pro jejich schopnost poskytovat na výstupu konzistentní produkt z hlediska kvality i navzdory nekonzistentnosti materiálu na vstupu. Další výhodou je úspora energie, kde ve srovnání s jinými možnostmi, jako například “fluid bed dryer”, tedy sušení ve fluidní vrstvě, vyžaduje méně energie a umožňuje energeticky ušetřit i při běhu na snížených kapacitách. Předností rotační sušárny je i možnost její modifikace dle potřeb a vlastností vstupního materiálu. Účelem možných modifikací je především optimalizace výrobního procesu. [3].

Po vysvětlení předností a obecných skutečností ohledně rotačních sušáren je na čase rozebrat funkci sušárny využití v projektu. Z důvodu obchodního tajemství však nelze k této práci přiložit podklady obdržené od zákazníka bez jeho písemného souhlasu. S ohledem na tento fakt bude všechn následující popis vycházet z daných podkladů, avšak pro názornost a demonstrativní účely bude využito principiální schéma, které se vůči originálu liší minimálně a je znázorněno na obrázku 2 níže.

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE



Obr 2: Principiální schéma rotační sušárny vápence [5]

Výrobní linka, nebo spíše její část zabývající se sušením vápence, lze principiálně rozdělit na čtyři hlavní sekce, a to:

- Přísun materiálu:  
Zprostředkovává dopravu a dávkování materiálu do sušícího bubnu.
- Sušící bubna se spalovací komorou:  
Jedná se o rotační bubna, ve kterém je přichozí materiál uveden do rotačního pohybu a sušen vystavením tepla přichozímu ze spalovací komory.
- Filtrace prachu (na obrázku navazuje na výstup exhaust air):  
Aby nedocházelo k hromadění prachu v komoře rotačního bubnu, je nutné, aby byl v provozu ventilátor, který za stejného principu jako běžný vysavač natáhne prach z bubnu do předehřáté násypky. Z násypky posléze prach projde filtrem a přidá se k již usušenému materiálu z výstupu sušárny.
- Výstup vysušeného materiálu:  
Během procesu sušení vystupuje z bubnu kontinuálně již vysušený materiál, který je třeba dál přepravit. V případě technologie využití v tomto projektu dochází k výstupu materiálu z bubnu i po nějaký časový interval po předání pokynu k zastavení linky.

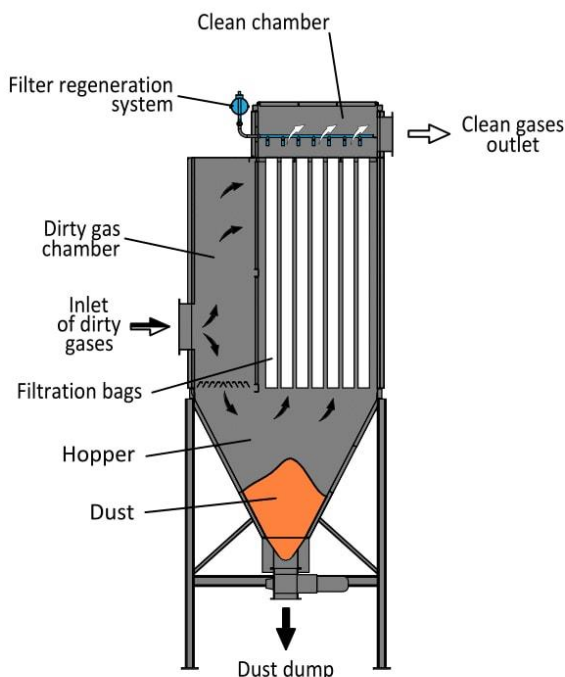
### 1.1.3 Princip funkce filtru

Filtrační jednotka je pro výrobní linku dodaná jiným výrobcem než rotační sušárna a nebyly k ní bohužel obdrženy dodatečné podklady krom soupisu s pokyny k řízení. Je proto výrazně komplikovanější popsat její funkci čtenáři i díky tomu, že není zcela jasné, o jaký typ filtru se přesně jedná.

Z ovládacích pokynů, rozmístění a názvů řízených komponent jako jsou motory, řídicí jednotky a tak dále v technologickém schématu lze říct, že se nejspíše jedná o filtr buď s horizontálními,

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

nebo vertikálními pytlými filtry. Pro názornost a větší shodu bude popis proveden na schématu vertikálního pytlového filtru uvedeném níže na obrázku číslo 3.



Obr. 3: Principiální schéma vertikálního pytlového filtru [6]

Princip funkce spočívá v tom, že vzduch znečištěný prachem prochází po vstupu do filtru přes vnější stranu pytlových filtrů, během čehož dojde k oddělení většiny kontaminantů ze vstupního plynu. Odloučený prach posléze spadne do násypky filtrační jednotky a může být přepraven k dalšímu zpracování se zbytkem materiálu z výstupu sušárny. [6]

V průběhu filtračních cyklů však nevyhnutelně dochází i k zanášení filtračních váček prachem odděleným ze vzduchu. Aby se zabránilo postupnému ucpávání filtru, je třeba zajistit jeho periodické čištění, a to buď za provozu, nebo během odstávek v případě dvou blokového filtru [6]. V případě jak filtru znázorněném na obrázku 3, tak i filtru využitém v projektu se jedná o očistu za provozu s využitím regeneračního systému taktéž k vidění na obrázku 3. Regenerace je realizována pomocí trysek aplikujících neznečištěný stlačený vzduch na filtrační váčky, čímž zajistí i jejich očistu od usazeného prachu. [6].

V případě projektu pro sušárnu vápence týkajícího se této bakalářské práce je ve filtrační jednotce dodatečně i ohřev násypky. Přídavný výhřev se promítá do technologického řízení tak, že je nutné podle výrobce zajistit jeho nepřetržitou funkci v průběhu operace, a že musí běžet alespoň dvě hodiny před spuštěním technologie. Jeho význam je stejný jako v případě výstupu ohřátého vzduchu z bubny sušárny, a to zajištění, že teplota stěn, resp. plynu neklesne pod rosný bod vody v extrahovaném vzduchu. Pokud by totiž došlo k poklesu teploty pod zmíněný rosný bod, došlo by ke kondenzaci vodních par na stěnách násypky a k opětovnému smíchání vody s nyní již vysušeným prachem materiálu.

## 1.2. TIA Portál

TIA Portál (Totally Integrated Automation portal) je plně integrované vývojové automatizační prostředí od firmy Siemens. Příkladem integrovaného softwaru je například zejména SIMATIC Step 7 a SIMATIC WinCC. [7]

- STEP7 v TIA portálu je inženýrský software sloužící ke konfiguraci rodin řídicích jednotek SIMATIC s7-1200, s7-1500 a s7-300/400. [7]
- WinCC slouží v TIA portálu ke konfiguraci panelů SIMATIC, průmyslových počítačů SIMATIC a běžných počítačů s profesionálním vizualizačním softwarem WinCC Runtime Advanced. [7]

Následujících několik sekcí teoretického rozboru okrajově seznamuje čtenáře s pojmy vyskytujícími se v této práci, které se týkají řídicí jednotky SIMATIC s7-1500. Ačkoliv náplní bakalářské práce je vizualizace výrobního procesu, je stále nezbytné realizovat i řídicí program, jelikož je s vizualizací přímo spojen. Následující sekce zmiňují tedy zejména strukturu vzniku a ukládání dat v TIA Portálu s využitím programovatelného automatu typu SIMATIC S7-1500.

### 1.2.1 PLC

PLC (Programmable Logic Controller) se do češtiny překládá jako programovatelný logický automat. V průběhu bakalářské práce bude rozebrána struktura řídicího programu, a následně využíváno propojení mezi vizualizací na operátorském panelu a řídicím programem v PLC typu SIMATIC S7-1500 od firmy Siemens.

### 1.2.2 PLC Tagy




Tag je v terminologii PLC označení pro unikátní identifikátor elementu v databázi, který se překládá na konkrétní vnitřní adresu dle typu zařízení.

U projektu ve vývojovém prostředí TIA Portálu jsou tagy seskupeny v tzv. tagových tabulkách. Jde o list obsahující soupis definic jednotlivých PLC tagů, kde je pro každý tag nezbytné, aby měl symbolický název spojený s konkrétní adresou a datovým typem. [8]

V TIA Portálu jsou rozlišovány tři adresní prostory, konkrétně:

- I – paměť sloužící k uchování obrazu vstupních proměnných
- Q – paměť sloužící k přiřazení hodnot výstupních proměnných
- M – adresní prostor pro vnitřní paměť programu [8]

Pro jednotlivé PLC tagy lze nastavit i dodatečné parametry, jako například atribut „retain“, který určuje, zda je hodnota daného tagu uchována v případě přerušení napájení řídicí jednotky. Na obrázku 4 níže je k vidění náhled struktury tagového listu. [8]

	Name	Data type	Address	Retain
	Motor1	Bool	%Q3.1	<input type="checkbox"/>
	Motor2	Bool	%Q3.2	<input type="checkbox"/>
	Control	Bool	%I3.3	<input type="checkbox"/>

Obr. 4: Definice PLC tagů pro TIA Portál [8]

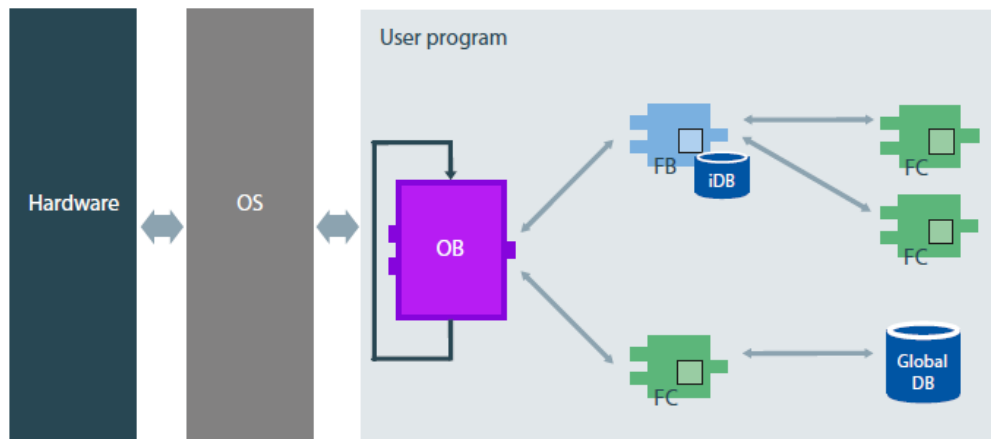


### 1.2.3 Organizační bloky v PLC

Organizační bloky (OB) tvoří rozhraní mezi operačním systémem (OS) a uživatelským programem. Operační systém je vyvolává například pro následující operace: [7][8][9]

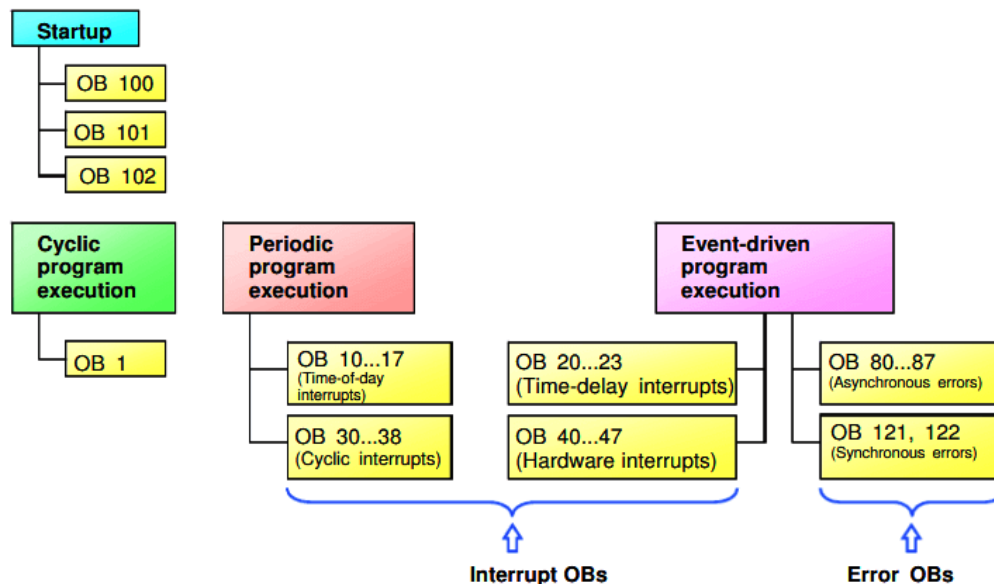
- Operace vykonávané při spuštění automatizačního systému
- Cyklické zpracovávání programu
- Vykonávání programu v instancích přerušení
- Ošetření chyb

Struktura interakcí mezi systémem a jednotlivými bloky je k vidění na obrázku 5 níže. Zkratky FB, FC a DB znamenají popořadě funkční blok, funkce a datový blok a jejich funkce a struktura je blíže vysvětlena v následujících kapitolách.



Obr. 5: Vzájemné interakce systémů a bloků uživatelského programu [8]

Organizačních bloků existuje několik předdefinovaných typů, jak je k vidění na obrázku 6 níže.



Obr. 6: Struktura a typy předdefinovaných organizačních bloků [9]

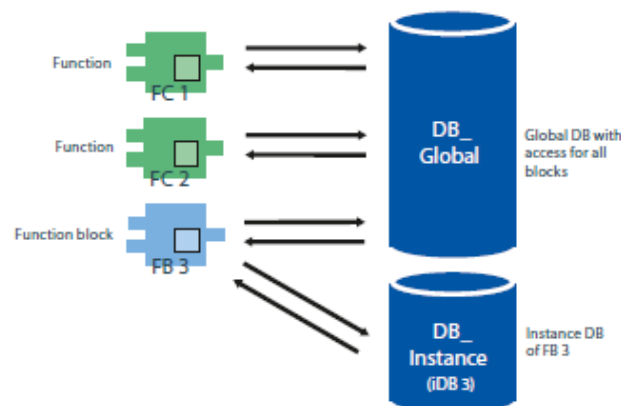
V projektu bude mezi využitě OB patřit OB 1, tedy základní hlavní cyklus celého programu značený názvem „Main“, který je vytvořen automaticky hned po přidání řídicí jednotky do projektu. Dalším využitým blokem je OB 100 pro inicializaci po spuštění PLC. Poslední využitě

OB bude OB 30, neboli cyklické přerušení s výchozí periodou 100 ms pro zajištění funkce regulátoru v pevně stanovených cyklech.

### 1.2.4 Datové bloky

Datové bloky (DB) jsou využívány k ukládání programových dat, ke kterým lze zároveň přistoupit i z ostatních bloků (např. funkčních bloků, které definují něco jako třídu objektu, ze kterých lze vytvářet instance a které budou vysvětleny v další sekci). [7]

Existují dva typy datových bloků, a to globální a instanční. Hlavním rozdílem je to, že instanční datový blok je vždy vztažen k nějakému funkčnímu bloku. Díky vazbě na funkční blok je struktura instančního DB pevně dána a nelze individuálně změnit, kdežto u globálního datového bloku si proměnné a strukturu programátor vytváří sám pro každý datový blok. [7] [8] Dělbou na globální a instanční bloky se v podstatě jedná o rozsah možného využití datových bloků v rámci programu. Instanční datové bloky pak vystupují pouze jako lokální k ukládání a poskytování dat dané instance provázaného funkčního bloku. Lze k nim sice stále přistupovat a zapisovat do nich i ze zbytku programu, ale drží pouze data týkající se příslušné instance, která je neustále modifikuje. Jednoduché schéma znázorňující rozsah využití typů datových bloků je na obrázku číslo 7.

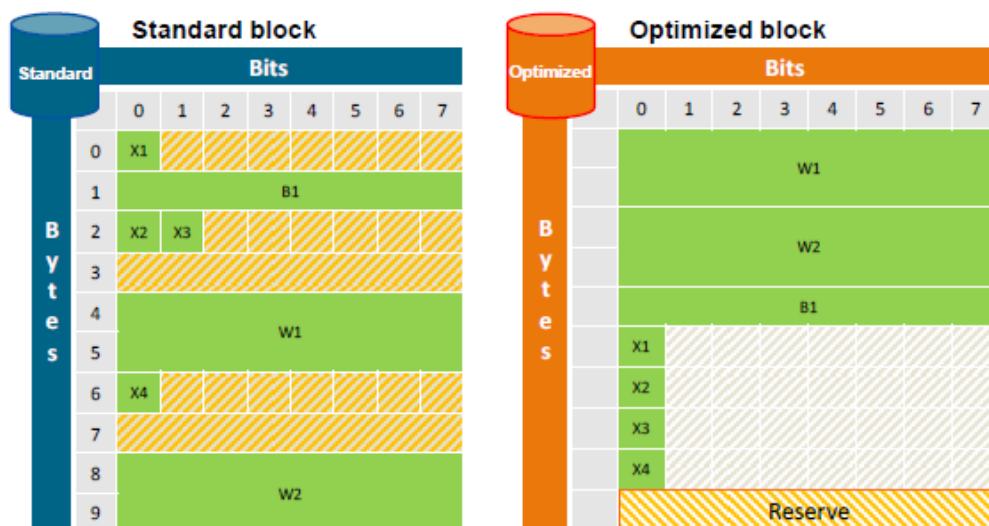


Obr. 7: Rozsah využití datových bloků v PLC [7]

Datové bloky dále rozlišují dva typy přístupu z hlediska uspořádání paměti, a to standardní a optimalizovaný. Bloky s optimalizovaným přístupem nemají fixně definovanou strukturu, a tedy u deklarace jednotlivá data nemají pevně definovanou adresu, ale pouze symbolický název. V případě neoptimalizovaného přístupu mají datové bloky pevnou strukturu a datové elementy při deklaraci dostávají jak symbolický název, tak i adresu. [7]

Optimalizované datové bloky volí rozdílný přístup v závislosti na verzi PLC. V případě řídicí jednotky typu SIMATIC s7-1200 probíhá optimalizace z hlediska využití paměti, avšak u rodiny SIMATIC s7-1500 je optimalizace zaměřena na rychlost přístupu k datům (liší se i formátem ukládání dat do paměti, jak bude zmíněno později). U PLC typu SIMATIC s7-1500 jsou data v paměti organizována tak, aby z nich procesor mohl číst a zapisovat do nich pomocí jedné strojové instrukce. Jednotlivé bity jsou proto ukládány po bytech, aby procesor nemusel maskovat přístup k jednotlivým bitům. [8] Níže na obrázku 8 je k nalezení srovnání využití paměti mezi optimalizovanými a standardními datovými bloky pro řídicí jednotku s7-1500.

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE



Obr. 8: Znázornění rozdílu ukládání do paměti mezi standardním a optimalizovaným blokem pro PLC s7-1500 [8]

Maximální velikost datových bloků je 64 kB, případně 16 MB, pokud se jedná o blok optimalizovaný. [8]

V rámci práce se využívá zejména neoptimalizovaný přístup, a to jak kvůli již existujícímu standardu funkčních bloků, tak i kvůli absolutní adresaci při přístupu vizualizačních objektů k vybraným datům z PLC.

### 1.2.5 Funkce

Funkce jsou bloky kódu bez vnitřní paměti. Všechny potřebné parametry tedy musí být definovány a předány funkci při volání. [7]

### 1.2.6 Funkční bloky

Funkční bloky (FB) jsou bloky s paměti, u kterých jsou jejich data napříč cykly ukládána v instancích datových bloků. FB umožňují použití jak statické, tak i dočasné proměnné. Rozdíl mezi nimi je takový, že dočasné proměnné, jak název napovídá, mají platnost pouze jeden cyklus a je nutné zajistit před každým jejich čtením i jejich inicializaci. Jinak totiž nejde zajistit předvídatelnou funkci programu. Statické proměnné však na rozdíl od těch dočasných svou hodnotu napříč cykly zachovávají a jsou tedy i hlavním důvodem, proč jsou následně data funkčního bloku uchovávána v instančních datových blocích. [7][8]

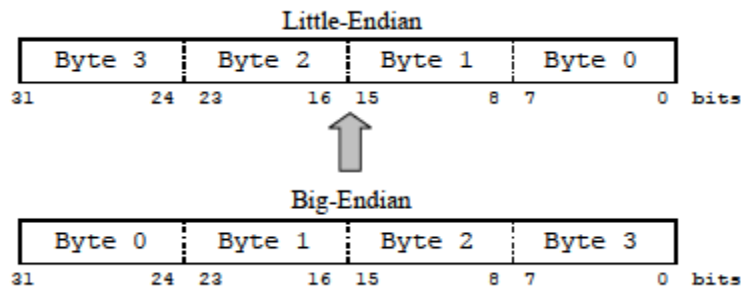
Funkční bloky obsahují bloky kódu, které se vykonají při každém cyklu a lze je volat opakovaně. Každé toto volání se z hlediska programového cyklu nazývá instance funkčního bloku a potřebuje vždy definovat i příslušný instanční datový blok. Do něj pak budou ukládány data náležící příslušné instanci. [7][8]

### 1.2.7 Formát ukládání dat do paměti

V rámci projektu se využívá přístup na absolutní adresu dat z později vysvětlených důvodů. Odkazování na absolutní adresy datových bloků vyzdvihuje otázku formátu dat uložených do paměti, kde se obecně nejčastěji využívá buď přístup big endian nebo little endian.

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

Endianita určuje pořadí ukládání bytů dat do paměti. Big endian ukládá nejvýznamnější hodnotu v datech na nejnižší adresu v paměti, a naopak little endian ukládá nejvýznamnější hodnotu v datech na nejvyšší adresu v paměti, což je i znázorněno na obrázku číslo 9. [10][11].



Obr. 9: Ukládání 32bitového čísla do paměti [11]

Například pro absolutní přístup na nultý bit nultého bytu v případě formátu big endian je potřeba se dotazovat na 24. bit uloženého 32bitového slova.

Pro řídicí jednotky od firmy Siemens SIMATIC s7-1500 se jedná o první řadu, kde procesorová architektura využívá 32bitový little endian. Standardní, tedy neoptimalizované datové bloky jsou u ní spolu s řadou SIMATIC s7-1200 k dispozici pouze z důvodů zpětné kompatibility s řadami SIMATIC s7-200, s7-300 a s7-400. Díky výše zmíněné zpětné kompatibilitě jsou data u neoptimalizovaných datových bloků ukládána ve formátu big endian a při zpracování v případě řídicí jednotky SIMATIC s7-1500 převedena do formátu little endian. Při využití optimalizovaného datového bloku jsou data již rovnou uložena ve formátu little endian a není tedy při zpracování zapotřebí žádné konverze. Data se z PLC do HMI zařízení přenáší tak, jak jsou uložena v paměti, tedy v případě neoptimalizovaného bloku jako big endian.[8]

### 1.2.8 Instrukce Cont\_C

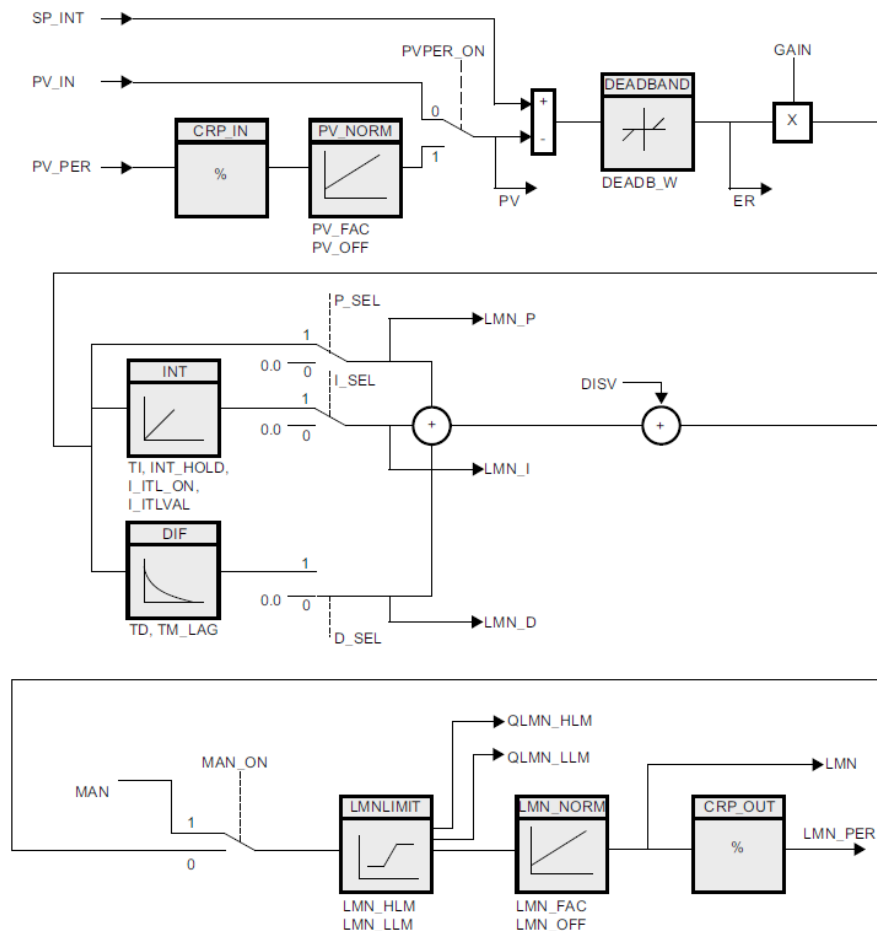
V pozdějších kapitolách práce přijde řeč na požadované podmínky regulace implementací propojení řídicích a řízených veličin. Regulátory, konkrétně proporcionálně integrační (PI), budou implementovány právě s využitím zde rozebrané instrukce Cont\_C v kombinaci s firemně standardizovaným blokem zajišťujícím rozhraní mezi firemním standardem a instrukcí. Přidaný blok dále ovládá konkrétní konfiguraci vstupů instrukce na základě svých nastavených parametrů.

Cont\_C je instrukce sloužící k řízení technických procesů se spojitým vstupem a výstupem. Původně byla na platformách SIMATIC s7-300, s7-400 a byla posléze přenesena i na platformu SIMATIC s7-1500. Krom využití přístupu optimalizovaných datových bloků a zachování parametrů po vypnutí PLC se instrukce pro řídicí jednotku SIMATIC s7-1500 vůči zmíněné dřívější verzi vůbec neliší. [7]

Prostřednictvím oné instrukce lze realizovat regulátory typu P, I, PI, PD a PID, kde D značí derivační složku, kterou nelze aplikovat samostatně. Pro správnou funkci výpočtu hodnot v regulátoru je třeba zajistit, že bude instrukce volána v pravidelných intervalech, nejlépe tedy z OB cyklického přerušení. Interval volání cyklického přerušení vykonávající tuto instrukci slouží jako vstup parametru CYCLE. [7]

Mimo možnost výběru typu regulátoru a regulačních konstant má instrukce jako vstupní argumenty i pásmo necitlivosti, limity rozsahu akčního zásahu, výběr automatického či manuálního módu, nastavení požadované hodnoty a možnost čtení vstupní hodnoty z programu, nebo rovnou z periferie. [7] Blokové schéma instrukce je k nalezení níže na obrázku číslo 10.

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE



Obr. 10: Blokový diagram funkce Cont\_C [7]

### 1.3. Vizualizační prostředí WinCC

#### 1.3.1 Popis HMI

Rozhraní mezi člověkem a strojem, nebo též ve zkratce HMI, je platforma/princip věnovaný propojení člověka, často operátora, se strojem, systémem, či zařízením. [12][13]

S využitím HMI lze například zajistit vizualizaci výrobního procesu, zobrazení jeho stavu a ovládání systému operátorem přímo z panelu. Jedná se tedy o poměrně široký pojem, který se aplikuje ať už na vestavěné obrazovky přímo na stroji, obrazovky počítačů poskytující informace o nějakém procesu, ale i třeba na mobilní telefon, který poskytuje možnost si vzdáleně zobrazit vybraná data z výrobního procesu. Myšlenka však zůstává všude stejná, a to poskytnout uživateli vhled do výrobního procesu a případně mu umožnit ho z části ovládat či korigovat. [12]

Rozhraní mezi člověkem a strojem se často používá v souvislosti s programovatelnými logickými automaty PLC, které řídí samotný výrobní proces. Právě komunikaci mezi PLC a operátorem zajišťuje HMI rozhraní. Toto rozhraní pak poskytuje operátorovi přístup k datům z PLC, a to jak v původním formátu, tak třeba i zpracované, například v grafu. Princip HMI zároveň vytváří i cestu, jak modifikovat data, která jsou nahraná v samotném programovatelném automatu, čímž lze například vybrat požadovanou recepturu, nebo upravit cílové hodnoty regulace.[12][13]

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

V rámci standardu pro SCADA<sup>1</sup> a automatizační systémy z roku 2008 je soupis minimálních náležitostí, které by měly být obsaženy v rámci vizualizačních obrazovek HMI. Standard bude pro účely práce sloužit spíše jako návod co se obsahu obrazovek týče. Obrazovky by tedy měly navrhnout výrobní proces obsahovat systém menu, který umožňuje jednoduchou navigaci mezi jednotlivými obrazovkami. Dále by měla existovat obrazovka zobrazující alarmové hlášení spolu s historií vyvolaných alarmů, a soubor obrazovek věnovaných elektrickému napájení a jiným fyzickým zařízením. Vizualizace analogových zařízení by měla obsahovat okna umožňující zobrazení trendu naměřených hodnot. [13]

### 1.3.2 SIMATIC HMI Unified Control Panel

SIMATIC HMI Unified control panel je verze operátorských dotykových panelů nabízených firmou Siemens. Poskytují propojení s vizualizačním softwarem SIMATIC WinCC Unified a oproti předchozím řadám panelů i výrazně zvýšenou výkonost [14]. V projektu je využívána verze MTP1500<sup>2</sup>, která má displej o úhlopříčce 15.6 palců s náhledem na obrázku číslo 11.



Obr. 11: Operátorský panel SIMATIC HMI Unified Comfort [15]

### 1.3.3 HMI tagy

Přístup k datům uložených/využívaných v PLC z HMI panelu je zprostředkovaný pomocí HMI tagů, které mohou být dvojího typu, a to buď interní, nebo externí. Stejně jako v případě PLC tagů jsou i HMI tagy v prostředí TIA Portálu ukládány v tagových listech. V případě, že se jedná o propojovací tag, je třeba ho navázat na požadovanou položku v paměti PLC. Je tedy takto možné číst vstupní, výstupní nebo i vnitřní proměnné v datových blocích, nebo v tagových listech uložených v paměti PLC. Externí HMI tagy jsou propojené s daty v PLC a je možné pomocí nich číst či přepisovat procesní hodnotu v paměti PLC programu. [16]

Externí tagy s pomocí nakonfigurovaných vlastností definují spojení s řídicím systémem a naváží výměnu dat. Vyměňovaná data jsou obrazem definované oblasti v paměti PLC, do které je přístup ke čtení i zápisu jak pro PLC, tak i pro HMI zařízení. [17][16]

Kromě základních datových typů tagů PLC je možné pomocí HMI tagů přistoupit i na instance datových bloků, či uživatelsky definované typy (UDT). Přístup k položkám v paměti PLC pomocí externích tagů může být referován buď symbolicky, nebo absolutně. V případě symbolického

<sup>1</sup> SCADA = Supervisory Control And Data Acquisition. V překladu „dispečerské řízení a sběr dat“

<sup>2</sup> (technické informace k dispozici na: <https://support.industry.siemens.com/cs/pd/1430614?ptdi=td&dl=en&lc=en-WW>).

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

přístupu stačí propojit pouze název cílového tagu z nabídky a zároveň dojde k propojení korespondujícího datového typu. Symbolické adresování je jediná možnost v případě, že by se u cílové položky jednalo o optimalizovaný datový blok. Absolutní adresování nabízí zejména možnost multiplexování tagů, kde cílovou část adresy, např. jako bit nebo číslo datového bloku lze propsat s využitím jiného tagu a použít ho jako placeholder. Pod pojmem placeholder je myšlen zástupný symbol/tag, který se za běhu programu dodefinuje, či může měnit svoji hodnotu. Pomocí využití multiplexování lze pak jedním<sup>3</sup> tagem přistoupit na velké množství paměťových lokalit. [16] Náhled deklarace multiplexovaného tagu v HMI tagovém listu je na obrázku 12.

HMI tags							
Name ▲	Data type	Connection	PLC name	PLC tag	Address	Access mode	Acquisition cycle
Motor_Object_Definition_DWord	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_Number].DBD136	<absolute access>	T1 s
Motor_Object_Status_DWord	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_Number].DBD76	<absolute access>	T1 s

Obr. 12: Deklarace multiplexovaného HMI tagu

V případě externích tagů, jsou jejich aktuální hodnoty přenášeny z PLC do procesní paměti HMI prostřednictvím komunikačního kanálu definovaného pomocí HMI propojení. Následně dojde k aktualizaci hodnoty těchto tagů v cyklických časech definovaných při jeho vytváření. Při přístupu k hodnotě tagu za běhu procesu, přistoupí WinCC k hodnotě požadovaného tagu, která byla načtena v předchozím cyklu. Může se tedy stát, že WinCC bude číst hodnotu, která je již v daném okamžiku jiná. [16]

Ve WinCC je limitován počet využitých HMI tagů na projekt na 8000 u panelů typu Unified. [18]

### 1.3.4 Textové listy

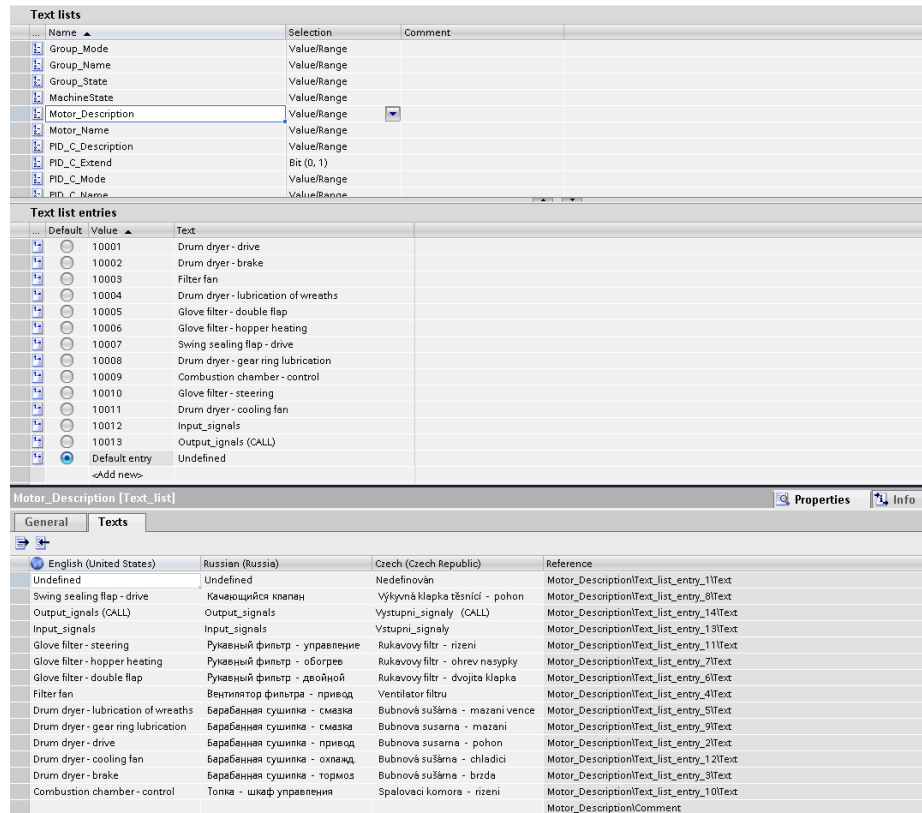
V projektu jsou veškeré texty ukládány do textových listů, které lze centrálně spravovat a udržovat jednotně pro všechny jazyky projektu v editoru projektových textů. Pro vizualizační účely lze však pomocí možnosti s názvem „textové a grafické listy“ vytvářet dynamicky se měnící textové a grafické položky, které lze posléze volat na vizualizačních obrazovkách. Právě na tuto možnost bude zaměřena současná a příští kapitola.

Každý textový list má své unikátní jméno a k jeho textovému obsahu lze přistupovat pomocí rozsahu hodnot, které má každý text přiřazené individuálně. Dojde-li k nastavení hodnoty v tomto rozsahu, zobrazí se posléze v nastavené destinaci příslušný text z daného textového listu. V případě změny jazyku projektu za chodu dojde k zobrazení textu přiřazeného danému jazyku na stejné číselné pozici. Pro potřebné úpravy, přidání překladu, či pouze nahrání potřebného textu je možné textové listy exportovat do formátu Office Open XML file a poté je opět naimportovat zpět. [7]

Existují dva typy textových listů, a to definovaný systémem anebo uživatelem. Uživatelsky definované textové listy mohou být libovolně vytvořeny, naplněny textem a mít libovolně definované hodnoty. U systémově tvořených listů nelze přidávat nové ani měnit hodnoty pro volání textu, avšak lze měnit text samotný. [7] Náhled struktury textového listu je uvedený na obrázku číslo 13.

<sup>3</sup>Jedná se sice o jeden multiplexovaný tag, ale k přístupu na různé adresy je třeba využít ještě jeden až dva další pomocné tagy udávající cílovou adresu přístupu.

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE



Obr. 13: Náhled tvorby textového listu pro vizualizaci později využitého v projektu

### 1.3.5 Grafické listy

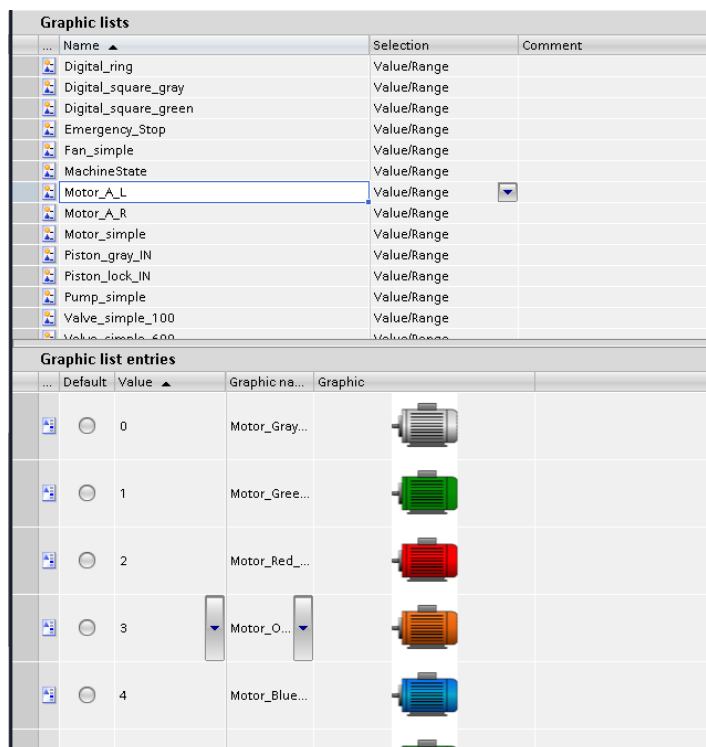
Během konfigurace lze vytvářet grafické listy s podobnou strukturou jako dříve zmíněné listy textové. Grafické ikony lze z těchto listů přiřadit například tlačítku, které má následně podle stavů měnit svou podobu, nebo pro vizualizační objekty „graphic I/O field“. V listech jsou opět jednotlivé ikony rozřazeny číselně, ať už individuálně či v číselných rozsazích a vybírá se z nich dle hodnoty propojovacího tagu. [7]

Listy mohou být opět multilingvální a v různých jazycích bez změny hodnoty propojovacího tagu měnit příslušné ikony. [7] Tato možnost se může hodit například pro zobrazení vlajek reprezentujících momentální nastavení jazyku projektu, nebo pokud by byl na grafických ikonách z nějakého důvodu potřeba text.

Grafické ikony lze vložit do listu buď z grafické knihovny projektu, nebo vložením nového obrázku v některém z následujících formátů: \*.bmp, \*.ico, \*.emf, \*.wmf, \*.gif, \*.tiff, \*.png, \*.svg, \*.jpeg a \*.jpg.[7] Náhled na tvorbu textového listu je k nalezení níže na obrázku 14.



## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE



Obr. 14: Náhled tvorby grafického listu pro vizualizaci později využitého v projektu

## 1.4. Komunikace

Komunikační kanál zajišťuje přenos dat mezi PLC a operátorským panelem. Jednou z možností propojení je využít propojovací technologii ProfiNet.

Základem pro všechny typy komunikací je konfigurace sítě, v jejím rámci se specifikuje propojení konfigurovaných zařízení. Zároveň je třeba zajistit, aby každé zařízení obsažené v síti mělo unikátní adresu a aby charakteristika přenosu mezi zařízeními byla konzistentní. [17]

### 1.4.1 ProfiBus

Profibus je mezinárodně standardizovaný fieldbus systém, což znamená, že je člen rodiny komunikačních protokolů určených pro průmyslovou aplikaci. Konkrétně se jedná o standard IEC 61158. [17][19]

Jeho charakteristickou vlastností je deterministický přístup ke sdílenému médiu, buď pomocí metody předávání tokenu mezi řídicími jednotkami, nebo přístupem master-slave v případě komunikace mezi řídicí jednotkou a koncovými zařízeními. [19]

Profibus je vhodný především ve výrobním průmyslu díky rychlosti odezvy. [17]

### 1.4.2 ProfiNet

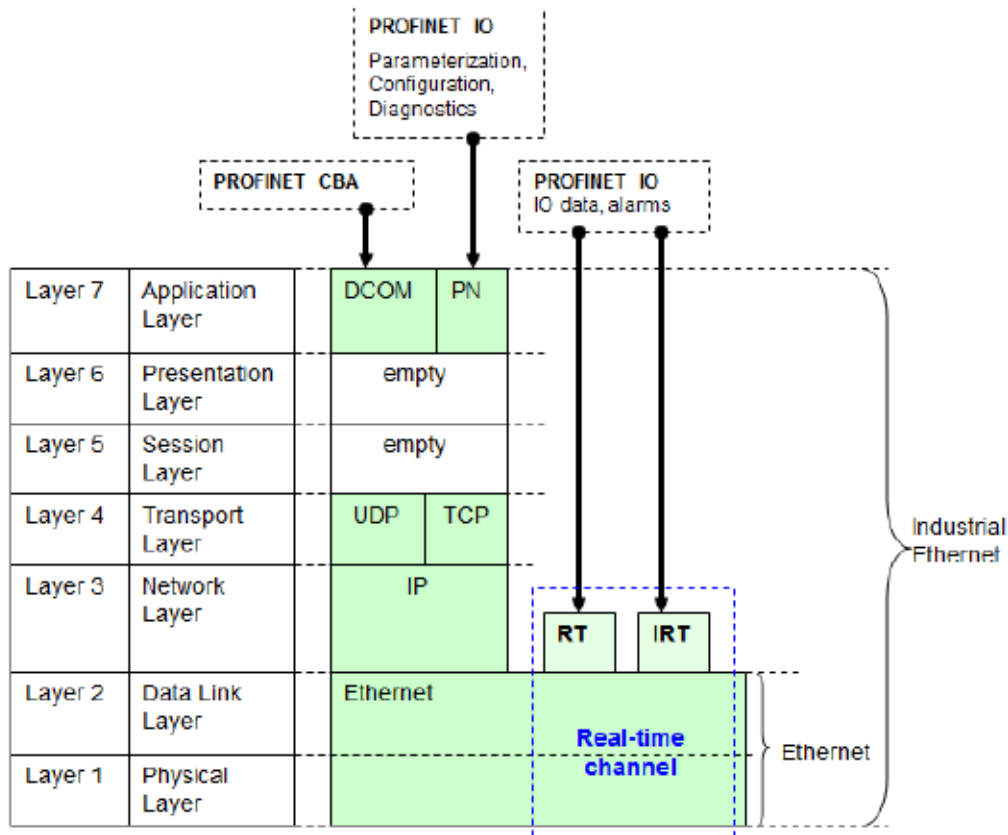
ProfiNet je otevřený standard průmyslového ethernetu pro automatizaci definovaný normou IEEE 61158. Využitím profiNetu lze realizovat vysoce výkonné automatizační řešení s nároky na přenos v reálném čase. [17][19]

ProfiNet je založen na průmyslovém ethernetu, který je vychází z klasického ethernetu. Průmyslový ethernet je nastavba/varianta ethernetu přizpůsobená pro průmyslové aplikace. Jako odlišnosti mezi průmyslovým ethernetem a klasickým ethernetem lze uvést například nižší přenosovou rychlost v řádu 100 Mb/s, existenci protokolu zajišťující přenos v reálném čase a

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

deterministický přenos. Navrch jsou v případě průmyslového ethernetu konektory a kabely přizpůsobené pro operaci v náročnějším prostředí. [19][20]

V nástavbě na průmyslovém ethernetu poskytuje profiNet integrovanou komunikaci přes fieldbus a ethernet a zároveň integraci fieldbus systémů jako například profiBus. [19] Na obrázku číslo 15 je k vidění přirovnání profiNetu k referenčnímu OSI modelu.



Obr. 15: Přirovnání průmyslového ethernetu a profiNetu k referenčnímu OSI modelu [19]

## 2. Praktická část

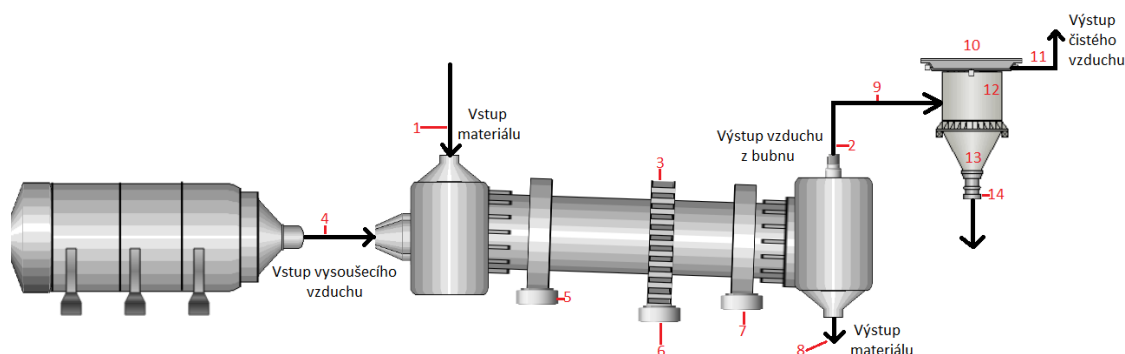
Po vysvětlení pojmů a nástrojů využívaných v práci přichází na řadu popis řešení zadaného projektu. Bude tedy v následných sekcích popsána struktura výrobní linky a dostupných technologických objektů. Následně se přikročí k popisu implementace řídicího programu linky na základě navržených programových vazeb a následně bude vytvořena i samotná vizualizace výrobního procesu sušárny vápence doprovázená o popis jejího návrhu a tvorby.

### 2.1. Struktura výrobní linky

Před implementací řídicího programu či vizualizačních obrazovek je v první řadě nutné seznámit se se strukturou samotné výrobní linky. Následující podsekcce tedy s využitím dříve provedeného teoretického rozboru výrobního procesu uvedou schéma linky s popisem dostupných technologických objektů a následně popíší strukturu řízení na základě provedeného návrhu doplněného o pokyny technologa a výrobců jak rotační sušárny, tak filtrační jednotky

#### 2.1.1 Analýza technologického schéma

Jak již bylo zmíněno v části teoretického rozboru, nelze k této práci přiložit podklady obdržené od zákazníka bez jeho písemného souhlasu. S ohledem na tento fakt bude všechen následující popis vycházet z daných podkladů, avšak pro názornost a demonstrativní účely bude využito schéma vytvořené z vizualizačních ikon navržených pro účely tohoto projektu s popisem tvorby v kapitole 2.3.6. Jde o schéma rotační sušárny spojené s filtrační jednotkou a je uvedeno na obrázku 16 níže.



Obr. 16: Popsané schéma linky sušárny vápence

Na obrázku 16 lze vidět referenční schéma rotační sušárny uvedené na obrázku 2 výše, avšak s přidaným popisem poloh signálů, na které se bude odkazovat současná kapitola o návrhu bloků. Body uvedené v obrázku mají následující význam:

1. Vstup materiálu k sušení – momentálně bez signálů, ale očekává se komunikace s předcházející technologickou částí. Zejména tedy pokyn ke startu a statusy o připravenosti a chodu.
2. Výstup vzduchu z bubny směřující do filtru – obsahuje dva senzory, a to teploty a tlaku výstupního vzduchu sušárny
3. Ozubený věnec – vztahuje se k němu řídicí jednotka zajišťující jeho mazání.
4. Vstup horkého vzduchu z pece do bubny – měří se zde tlak vstupního vzduchu do sušárny. Pro pec samotnou jsou pro náš program k dispozici pouze řídicí signály start/stop a signál k modulaci výkonu pece.

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

5. Levý podpěrný váleček – váleček, na kterém se nachází 4 senzory teploty k detekci tření, respektive kontrole potenciální poruchy ložiska.
6. Pohon rotačního bubnu spolu s brzdou pohonu, frekvenčním měničem a chladicí ventilátor pohonu. Dále je zde senzor teploty vynutí motoru.
7. Pravý podpěrný váleček – stejný jako bod pět, avšak na pravé straně.
8. Výstup materiálu z bubnu – měří se zde teplota výstupního materiálu. Očekává se zde ještě signál o připravenosti následující technologické části.
9. Vstup vzduchu do filtru – Zde je dodatečný senzor teploty na vstupu filtru. Mezi bodem 2 a 9 je ještě klapka připouštějící čerství vzduch do oběhu.
10. Regenerační jednotka filtru – zde je k dispozici řídicí jednotka regeneračního cyklu k očištění zanesených pytlových filtrů, a zároveň binární senzor tlaku vstupního stlačeného vzduchu.
11. Výstup čistého vzduchu z filtru – Vyfiltrovaný vzduch je natažen ventilátorem filtrační jednotky a následně odveden pryč. Zde mezi senzory patří dvě měření teploty ložisek ventilátoru a dva senzory vibrací.
12. Senzory hladiny materiálu v násypce – zde jsou dva hladinové binární senzory maximální hladiny. Jeden určuje překročení varovné hranice, druhý překročení kritické hranice. Proto jsou v projektu značeny jako vyšší a nižší maximální senzor hladiny násypky.
13. Ohřev násypky filtru – Zde je jednotka ohřevu řízená termostatem a binární senzor poklesu tlaku filtru
14. Výstup prachu z filtru – Vyfiltrovaný prach se zde přes dvojitou výkyvnou klapku filtru řízenou pohonem posílá na stejný dopravník jako výstupní materiál sušárny.

Z dříve zmíněného popisu technologického schéma sušárny vápence jsou v rámci této zakázky řízeny pouze partie týkající se bubnu rotační sušárny se spalovací komorou a filtrace prachu. Do zmíněných dvou skupin budou v rámci celé práce rozděleny jak technologické objekty, tak řídicí signály.

### 2.1.2 Popis řízení filtru

Pro sestavení struktury ovládání filtrační jednotky byl vytvořen prvotní návrh blokačních podmínek a technologických návazností na základě elektro dokumentace a technologického schéma výrobní linky. Tento návrh byl následně rozšířen, aby splňoval i požadavky výrobce použitého filtru, doplněn požadavky technologa a následně schválen zákazníkem. Výsledná implementovaná funkce je následující<sup>4</sup>:

- Pro startovací fázi je požadavek, že spouštění filtru může začít až po dvou hodinách ohřevu násypky. Po uplynutí minimálního intervalu ohřevu a pokynu ke startu má následně dojít ke spuštění dvojitě klapky filtru sloužící k přepravě separovaného prachu z násypky filtru. Po zapnutí klapky filtru a obdržení zpětné vazby o jeho funkci má dojít ke spuštění regeneračního časovače řídicí jednotky a až poté spustit ventilaci.
- Během operační fáze má být zajištěna nepřetržitá funkce transportu prachu z násypky a její ohřev řízen termostatem tak, aby teplota zdí neklesla pod rosný bod vody v extrahovaném plynu. Zároveň by měl být zajištěn operační bod chodu ventilátoru podle specifikovaných parametrů vstupní klapky.

---

<sup>4</sup> U všech bodů dochází samozřejmě i ke kontrole poruch a stavu signálů.

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

- Při standardním vypínání jednotky má dojít prvně k vypnutí ventilátoru a po jeho vypnutí a uplynutí jednoho regeneračního cyklu vypnout regenerační časovač (řídící jednotku). Po pěti minutách od vypnutí regeneračního systému je třeba vypnout i dvojitou klapku filtru.

Ke standardní vypínací sekvenci má dojít buď po pokynu operátora nebo po určité době od selhání regeneračního časovače, dosažení alarmující hladiny poklesu tlaku, selhání dvojitě klapky, nebo po aktivaci nižší vysoké hladiny prachu v násypce.

K okamžitému vypnutí bez ohledu na proceduru má dojít ve chvíli, kdy je dosažena poruchová hladina poklesu tlaku, nebo došlo k aktivaci horní krizové hladiny prachu v násypce filtru.

### 2.1.3 Popis řízení rotační sušárny

K řízení technologické části rotačního bubnu sušárny byl opět na základě elektro dokumentace a technologického schéma celého projektu učiněn prvotní návrh blokačních podmínek a návazností. Nástřel podmínek byl posléze doplněn a rozšířen po obdržení dodatečných podkladů, jako dokumentace od výrobce rotační sušárny a soupisu pokynů od technologa. Implementovaná funkce je následující:

- Ke spuštění skupiny technologického celku je nejprve nutné obdržet zpětnou hlášku o chodu filtrační části, a aby navazující linka byla připravena (zatím provizorně, neznáme signály navazující, ani předcházející technologie). Samozřejmě je zároveň pro spuštění potřeba, aby žádná z využívaných komponent nebyla v chybě, či senzory nedetekovaly kritickou hranici ať už teplot či tlaků. Pro sušárnu se nejprve spustí řídicí jednotka mazání ozubeného věnce, posléze odbrzdí a spustí pohon bubnu na minimální otáčky, které se po rozběhu budou postupně zvyšovat. Po aktivaci pohonu se zapne chladicí ventilátor motoru a řídicí jednotka spalovací komory. Jakmile je celá skupina aktivní a teplota výstupního vzduchu z bubnu ohřáta alespoň na teplotu 80 °C se pošle pokyn o připravenosti k přijetí materiálu na vstupu sušárny.
- Při vypínání skupiny nesmí dojít k aktivaci brzdy za chodu pohonu. Krom pokynu operátora, dojde k vypnutí v případě, kdy je překročena mezní teplota 90 °C na senzorech podpěrných válečků (což by implikovalo tření a zřejmě poškození ložisek), nedojde k promazávacímu cyklu v určitém intervalu, je překročena teplota vinutí motoru, nebo pokud bylo detekováno selhání senzoru či nějaké z komponent. Vypínání skupiny probíhá v opačném pořadí než u zapínání, s tím že po zastavení přísunu materiálu se počká před vypnutím spalovací komory 20 minut k dosušení veškerého materiálu. Po odstavení komory se pohon bubnu sušárny ponechá ještě 30 minut v provozu, aby vypadl všechny sušený materiál a buben částečně vychladl, s tím, že chladicí ventilátor pohonu sušárny se vypne až po zchlazení motoru pod stanovenou teplotu. Následně řídicí skupina po další prodlevě na dočištění od prachu vyšle pokyn k vypnutí filtrační části. Při nouzovém zastavení zapříčiněném poruchou odprašovacího zařízení nebo odvodu materiálu je od výrobce požadavek na technologické vypnutí pohonu bubnové sušárny, tedy tak jak bylo specifikované dříve. Výjimka nastává akorát v případě poruchy/vypnutí ventilace, a to v podobě okamžitého odstavení spalovací komory.

## 2.2. Návrh řídicího programu

Na popis struktury výrobní linky a dostupných technologických objektů naváže tato kapitola tvorbou řídicího programu projektu. Nejprve bude věnována pozornost přípravě projektu v podobě tvorby hardwarové sestavy a definici tagových listů projektu. Následně se bude

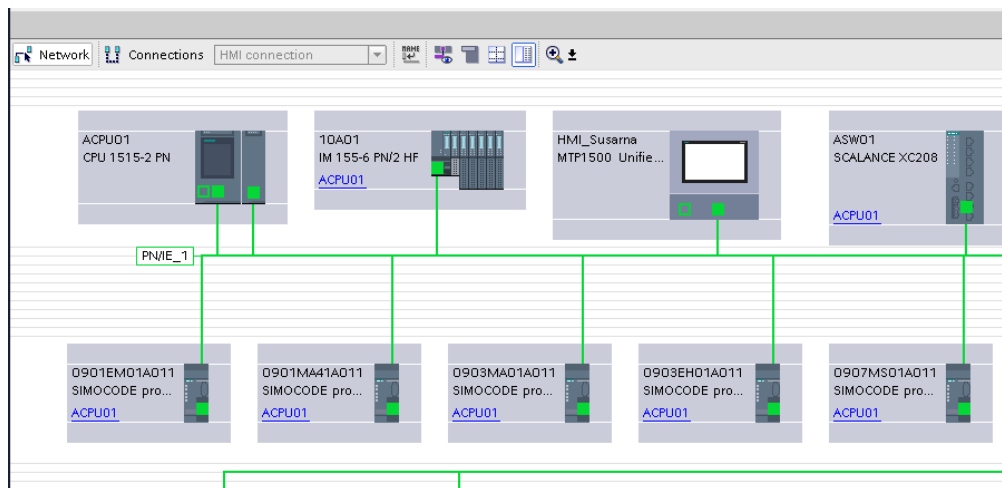
## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

pokračovat stručným popisem struktury využitého programového firemního standardu a jeho implementací do projektu. Závěrem kapitoly je již vysvětlení a demonstrace tvorby napojení programových vazeb zajišťujících dodržení popisu výrobního procesu zmíněného v předchozí kapitole.

### 2.2.1 Základní konfigurace projektu v TIA Portálu

Řídicí systém PLC řady S7-1500 od firmy Siemens, který je použit v tomto projektu, je programován ve vývojovém prostředí TIA Portal. V tomto prostředí se vytváří jak hardwarová sestava systému, tak i celý jeho program řízení. Navíc TIA Portal umožňuje do jednoho projektu zahrnout i program pro vizuální zařízení HMI, konfigurace měničů či jednotek soft startérů, pokud byly využity výrobky firmy Siemens.

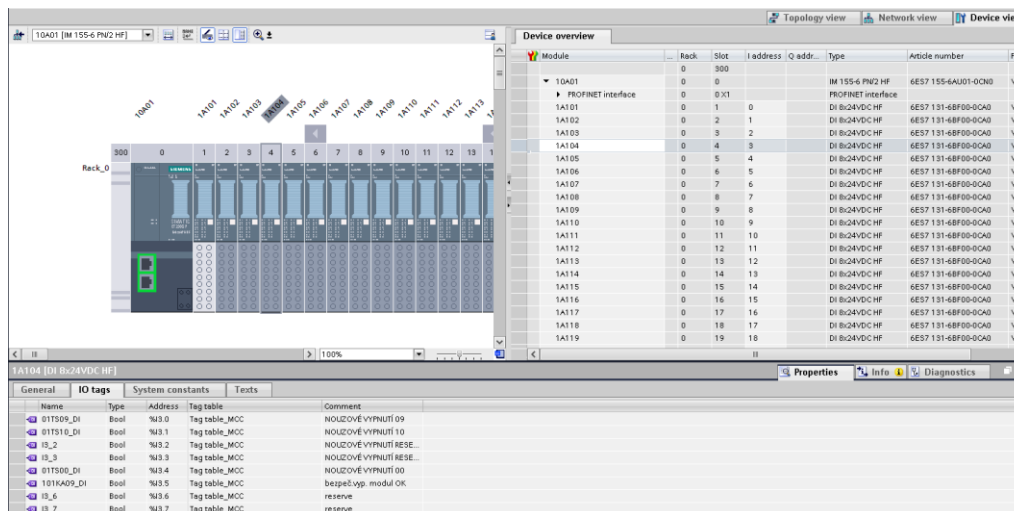
V hardwarové konfiguraci řídicího systému PLC se nakonfigurují všechna zařízení, se kterými bude PLC komunikovat. Jedná se tedy zejména o definování vzdálených periferií, se kterými si bude PLC cyklicky vyměňovat data. V případě tohoto projektu jde o distribuované vstupy/výstupy řady Simatic ET200SP, frekvenční měniče řady Sinamics a jednotky motorové ochrany řady Simocode. Je potřeba definovat zejména typ spojení (použitá komunikační sběrnice, v našem případě typ ProfiNet), typ komunikačního protokolu a vyčleněný adresní prostor pro výměnu cyklických dat (paměť vstupů a výstupů popsané výše). Obrázek propojení je k vidění níže na obrázku 17.



Obr. 17: Struktura propojení hardwarové sestavy projektu

Vzdálené periferie řady ET200SP je třeba v hardwarové konfiguraci nadefinovat dle zapojené sestavy vstupních a výstupních karet vycházející z elektro schématu a náhled lze vidět na obrázku 18.

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠARNY VÁPENCE



Obr. 18: Definice vstupních a výstupních signálů na kartách řídicí jednotky v editoru hardwarové konfigurace projektu

Po vytvoření hardwarové sestavy je ještě potřeba definovat list tagů obsahující vstupní a výstupní proměnné v PLC. Soupis těchto signálů vychází z elektro dokumentace projektu a lze z ní většinou získat přímo export těchto signálů do tabulky, čímž se tvorba tagů velice zjednoduší. Názvy a komentáře proměnných z tagových listů se v TIA Portálu vizuálně propíší až ke vstupním a výstupním kartám, jak lze taktéž vidět na obrázku z výše.

Výsledkem hardwarové konfigurace je, že řídicí systém PLC zná, s jakými periferiemi a jakým principem si s nimi vyměňuje data a tato data umí přiřadit k proměnným v programu.

## 2.2.2 Struktura využitého firemního programového standardu

Po vytvoření hardwarové konfigurace včetně listu tagů (signálů) se přistoupí k vytvoření vlastního řídicího programu. V tomto projektu byl využit firemní standard používaný jak pro řízení procesů (jako jsou např. vápenky a cementárny), tak pro FACTORY AUTOMATION (stroje a výrobní linky). V následujících odstavcích bude krátce popsána jeho struktura a implementace.

Základem firemního standardu je rozčlenit řízení na jednotlivé technologické objekty, které mají již své hotové funkční bloky a posléze programovat jen vazby mezi nimi. Mezi základní technologické objekty patří zejména motory (obecně pohony), ventily, písty, klapky, analogová měření či digitální senzory. Princip spočívá v tom, aby se s každým projektem nemuselo znovu programovat základní chování těchto objektů. Příkladem může být u pohonů hlídání jistištění, teplot vinutí, zpětného hlášení apod. Jelikož je firemní standard propojen i s vizualizačními objekty pro HMI, lze správným využitím takového standardu docílit rychlého naprogramování projektu do fáze ručního řízení.

Standard tedy využívá již připravených funkčních bloků, kde se pro každý objekt vytvoří adekvátní instance. Jelikož jsou tyto bloky napsány obecně, obsahují v sobě definice svého chování. Kombinacemi těchto definic lze docílit požadovaného chování jednotlivé instance objektu. Z tohoto důvodu se struktura programu dělí na dvě části, konkrétně:

- Inicializace technologických objektů:

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

Jedná se o funkci, ve které jsou pro každý objekt přiřazeny potřebné parametry, rozsahy a využívané vstupy potřebné k zajištění požadované funkce. Aby byly definované vlastnosti objektu spolehlivě zajištěny po celou dobu vykonávání programu, je funkce volána v každém cyklu hlavního programu.

- Volání technologických objektů:

Po dokončení inicializace je volána funkce, ve které jsou volány jednotlivé instance daného typu funkčního bloku. V tomto volání se k dané instanci přiřadí příslušné vstupní a výstupní signály. Funkční blok vstupní signály v kombinaci se svou vnitřní pamětí zpracuje a přiřadí na své výstupy hodnoty v závislosti na typu daného objektu. U motoru by se jednalo například o povely k dopřednému, či zpětnému chodu a světelnou signalizaci. V případě funkčního bloku obstarávajícího analogové měření by na výstupu zobrazil hodnotu měření přepočteného z periferií, nebo například signalizoval překročení povolených mezí měřené veličiny.

Dále jsou zde doprogramována propojení s ostatními instancemi a s nadřazenou logikou programu, aby se zajistili požadované automatické funkce a technologické blokace.

V rámci projektu budou využívány jak v programové, tak i vizualizační části následující objekty z firemního standardu:

- Objekt motoru – blok zprostředkovávající rozhraní mezi programovou logikou, vstupními signály a motorem samotným. Zajišťuje tedy vyhodnocování stavů motoru, poruch, a zda může předat z programu pokyn k chodu motoru. Blok zároveň rozlišuje ve vnitřní logice na základě toho, zda je v automatickém, manuálním či místním módu. Pro motory s frekvenčním měničem je přidán další funkční blok, který zajišťuje transformaci povelů od motoru do komunikačního protokolu výrobce daného frekvenčního měniče a na zpět.
- Objekt analogu – zprostředkovává zpracování vstupního analogového signálu ze senzorů a jeho přepočtení na stanovené rozsahy. Umožňuje kromě čtení měřené hodnoty i nastavení horních a dolních povolených mezí signálu pro tři různé hladiny. Hladiny jsou konkrétně varovací, alarmové a procesní a lze na jejich základě posléze v programu rozlišovat různé scénáře stavu linky.
- Objekt digitálu – zpracovává vstupní binární signál a na výstupu poskytuje buď jeho obraz, či inverzi v závislosti na nastavení inicializační funkce. Využívá se tedy pro jističe, tlačítka a případná měření kritických mezí, jako je například maximální výška hladiny v nádrži. Hlavním důvodem k jeho používání je standardizace a následné napojení na vizualizaci.
- Objekt skupiny – slouží jako platforma k propojení a řízení technologických celků. V bloku však není provázání řízených objektů zajištěno vnitřně a je tedy nutné posléze propojení vytvořit v rámci ovládaných částí. Funkce bloku se zaměřuje spíše na vyhodnocování stavů technologie a vydávání centrálních pokynů startu, chodu, zastavení, nouzového zastavení sekcí linky a kvitování chyb.
- Objekt navolení – slouží jako přepínač využíván zejména v kombinaci s HMI.

Některé standardizované objekty (v tomto projektu pouze objekt motoru) rozlišují tři provozní režimy. Konkrétně:

- Automatický – povely objekt dostává z logiky řídicího programu PLC
- Manuální – povely objekt dostává z operátorské stanice/panelu



- Lokální – povely objekt dostává z tlačítek na místní ovládací skříňce

V dalších kapitolách budou zmiňovány z hlediska standardu automatické povely objektů. Ty slouží k řízení objektů programem v automatickém režimu a jsou definovány ve funkci volání standardizovaných objektů, přičemž jejich logika se nakonfiguruje bokem a výsledek nakonec přivede na vstup daného instančního bloku. Dalším často zmiňovaným bodem budou blokační podmínky, které mají stejnou strukturu napojení jako automatické povaly. Blokace mohou být trojího typu, konkrétně:

- Startovací blokace se uplatňuje pouze při zapínání objektu. Jakmile je objekt již v chodu, blokace není uplatněna. Nevztahuje se na lokální režim.
- Provozní blokace je aktivní jak při zapínání, tak po celou dobu provozu. Nevztahuje se na lokální režim.
- Ochranná blokace je aktivní stejně jak provozní blokace po celou dobu provozu, avšak platí navíc i v lokálním režimu.

### 2.2.3 Implementace standardizovaných objektů do projektu

Jelikož již byl v předchozí kapitole vytvořen projekt, jeho hardwarová konfigurace a definován tagový list, lze pokročit tvorbou instancí funkčních bloků pro všechny technologické objekty využívané v projektu. Opět šlo o poměrně jednoduchý krok, kdy se obdržený soupis objektů porovnal s elektro dokumentací a posléze se každý vytvořil jako instance příslušného funkčního bloku, které se přiřadil název a číslo instančního datového bloku.

Parametry instancí funkčních bloků standardizovaných objektů je nyní potřeba nastavit v inicializační funkci. Základ inicializace je většinou stejný, a proto stačí pro každý objekt stejného typu nakopírovat výchozí definiční set a posléze individuálně upravit pár potřebných signálů v každém objektu. Tato dodatečná konfigurace se opírá o obdržené podklady a opět zejména elektro dokumentaci. Příkladem vnitřní definice, ve kterém se mohou lišit digitální objekty je invertovaný vstup. Důvodem pro jeho případné využití je často obrácená logika pro signály zajišťující bezpečnost funkce, jističe a vypínače.

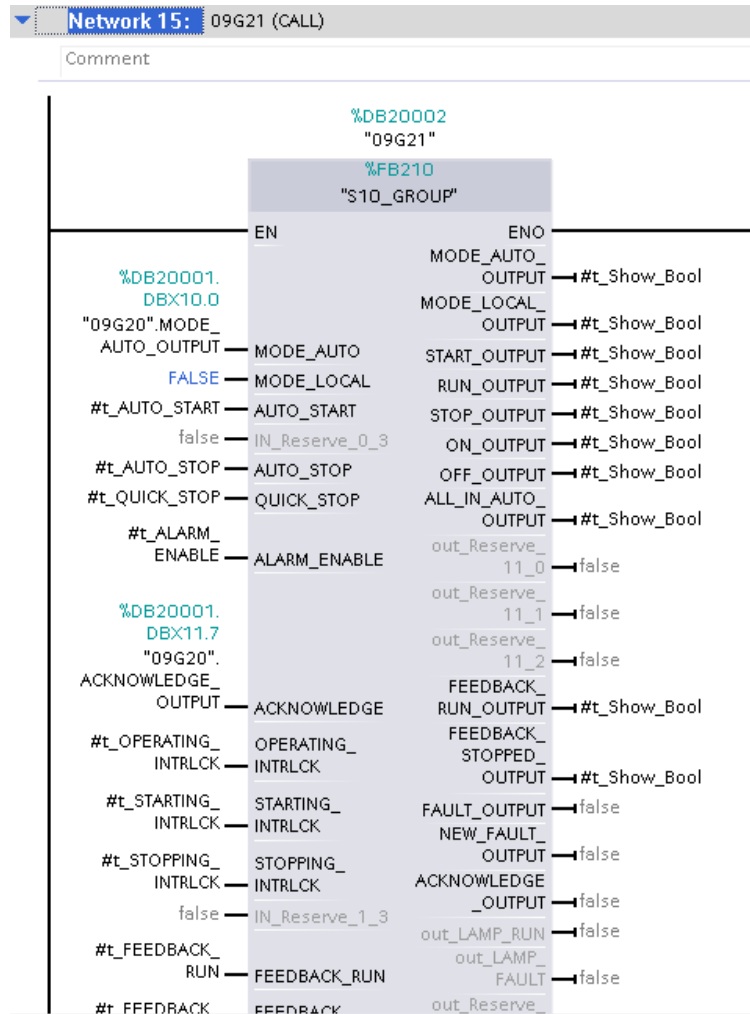
Tvorbu inicializace technologických objektů lze rovnou spojit s tvorbou jejich volání, a to ve smyslu, že vstupní signály, jako lokální řízení, zpětné hlásky z frekvenčních motorů a tak dále, se napojí na vstupy vytvořených instancí funkčních bloků.

### 2.2.4 Implementace programových vazeb

Po vysvětlení a implementaci využitého firemního standardu bude v této kapitole věnována pozornost vysvětlení struktury programu a implementace programových vazeb tak, jak byla dříve navržena dělba objektů do technologických skupin a jejich blokací. Tabulka vysvětlující názvy signálů uvedených na obrázcích této kapitoly je k dispozici v přílohách dokumentu.

Struktura programového řízení byla navržena tak, že obě dříve určené technologické skupiny budou v automatickém módu ovládány centrálně s využitím standardizovaného objektu skupiny. Objekt skupiny, jak již bylo vysvětleno v předchozí kapitole, slouží sice k řízení technologických celků, avšak navázání jednotlivých objektů musí být provedeno externě. Základ tohoto propojení se udělá navázáním řídicích signálů „MODE\_AUTO\_OUTPUT“ a „ACKNOWLEDGE\_OUTPUT“ z výstupu skupiny na vstupy řízených objektů. Jinými slovy, režim provozu technologických objektů skupiny je určen z dané řídicí skupiny, ze které lze rovněž kvitovat plošně chyby. Volání objektu skupiny je znázorněné na obrázku 19 níže.

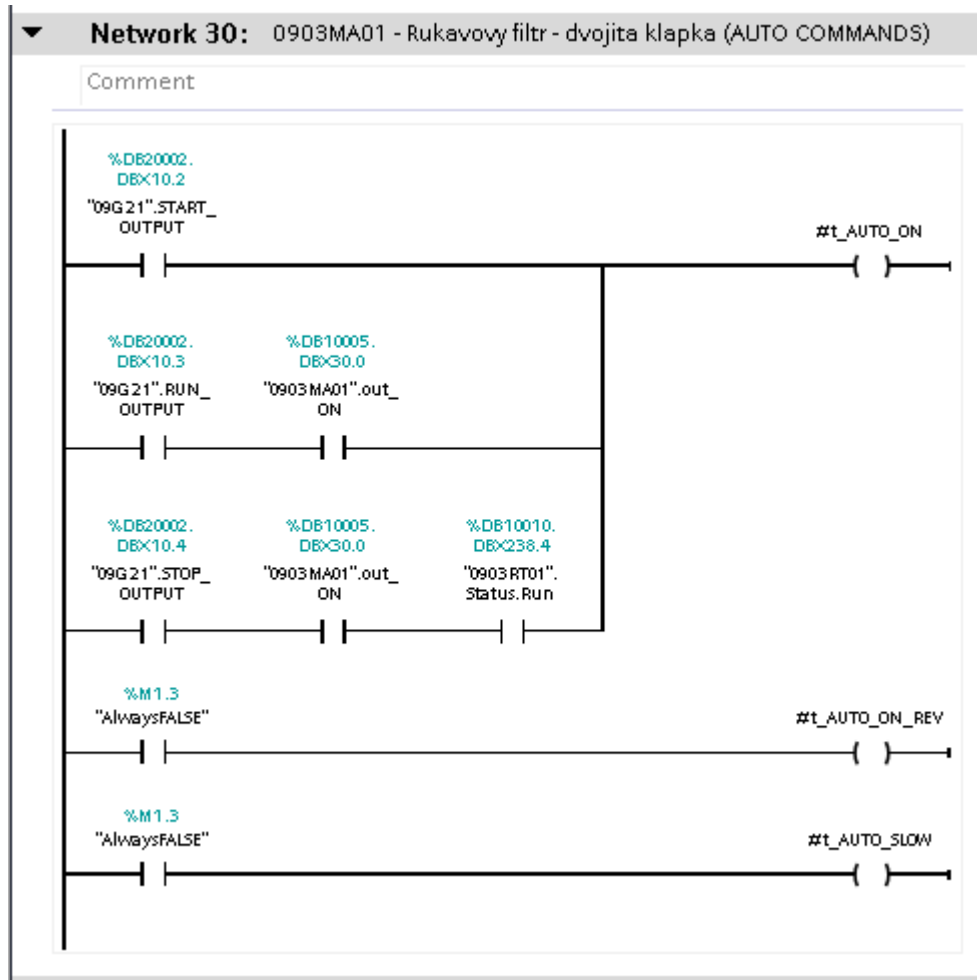
VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE



Obr. 19: Náhled volání objektu skupiny filtru

Po propojení režimu provozu objektů se skupinou, je potřeba rovněž navázat podmínky pro spuštění chodu řízených objektů. Výstupy skupiny Start, Run a Stop tedy spolu s dodatečnou logikou určují pořadí zapínání a vypínání pohonů právě díky dříve zmíněným automatickým pokynům například k chodu motoru. Zde jsou podmínky navázány tak, aby objekty byly technologicky zapínány a vypínány v pořadí dle podmínek stanovených v popisu řízení jednotlivých sekcí. Příkladem může být následující obrázek 20, na kterém je automatický pokyn k zapnutí dvojité klapky filtru. Řídící skupina po obdržení pokynu ke spuštění nejdříve provede signalizaci a nastaví výstupní signál Run. Po dokončení signalizace skupina nastaví signál Start, čímž dojde i k zapnutí dvojité klapky filtru, jelikož je vyjma ohřevu první v pořadí technologického zapínání filtru. Jakmile dojde k zapnutí všech řízených prvků a skupina obdrží zpětné hlášky o jejich chodu, změní výstupní signál Start na false a ponechá pouze signál Run. Po spuštění bude klapka v provozu do doby, než skupina přestane vysílat signál chodu. Tedy pokud buď nastala blokáce skupiny, nebo obdržela pokyn k vypnutí. Jakmile přijde od skupiny Stop (a tedy má dojít k technologickému vypnutí), bude klapka v provozu tak dlouho, dokud se před ní dle popisu procesu nevypne řídicí jednotka regeneračního cyklu filtru. Stejnou strukturu, jaká byla uvedena na příkladu, mají i všechny ostatní řízené objekty.

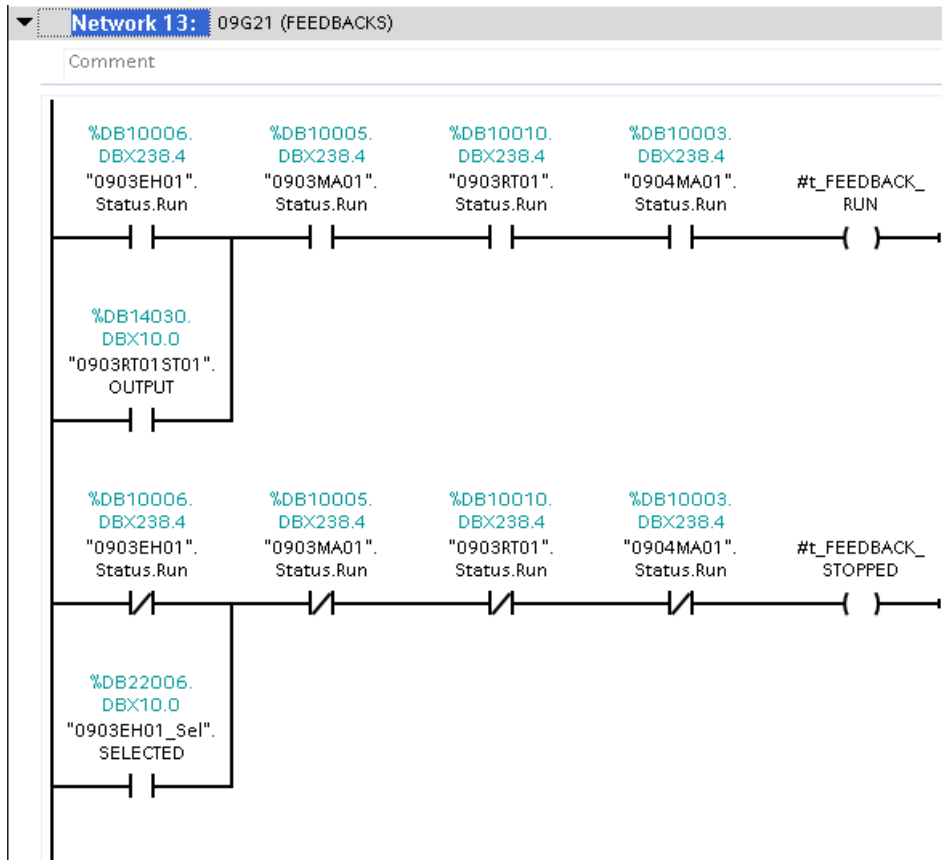
VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE



Obr. 20: Provázání pokynů ke spuštění pohonu dvojitě klapky filtru spolu s podmínkami jejího technologického vypnutí

Jak již bylo zmíněno dříve, je pro vyhodnocení ukončení startovací etapy a kontrole správné funkce potřebné pro skupinu zajistit zpětnou vazbu o chodu řízených komponent. Zpětné hlášení od objektů do skupiny lze udělat kaskádou podmínek o stavu chodu instancí funkčních bloků, jak lze vidět na obrázku 21. Na obrázku je ještě krom statusu chodu pohonu výjimka jak pro detekci zpětné vazby, tak pro detekci o jejím přerušení. Signál 0903RT01ST01 je signalizace z termostatu, během které není potřebné mít zapnutý ohřev násypky. Signál 0903EH01\_Sel zase označuje zaškrtnutou volbu operátora k funkci ohřevu bez ohledu na chod skupiny filtru.

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

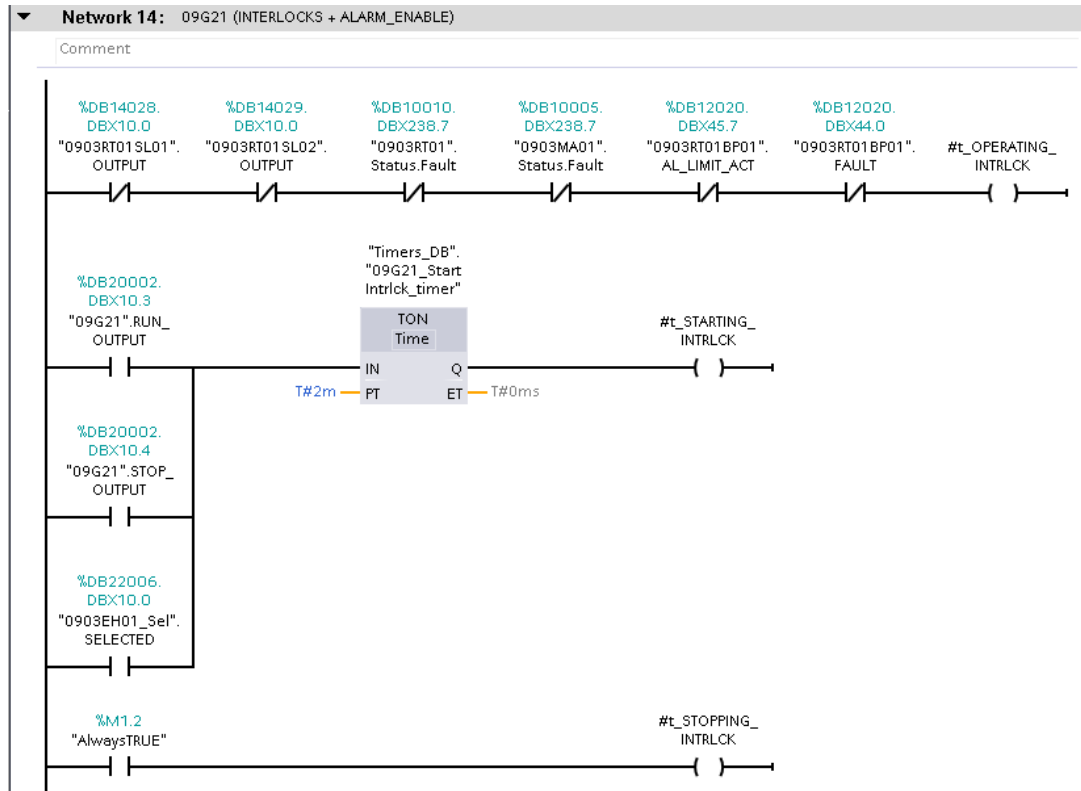


Obr. 21: Zpětné hlášení do skupiny o chodu řízených komponent

K zahrnutí situací reakcí na chyby a mezní hladiny měření jsou využity právě podmínky blokadí zmíněné u vysvětlení struktury firemního programového standardu. Podmínky blokadí se implementují do proměnných s názvem „interlock“, které mají tři různé typy. Pro implementaci blokadí programu jsou jisté odlišnosti v případě blokadí skupiny a blokadí objektu motoru. Prvním rozdílem, který lze uvést je absence ochranné blokace v případě skupiny, jelikož objekt skupiny funguje pouze v automatickém režimu. Druhý rozdíl je ten, že dojde-li k blokaci skupiny, nastaví výstup Stop do logické jedničky, kdežto všechny ostatní výstupy povelů do nuly, a tedy dojde k technologickému vypnutí sekce řízené danou skupinou. Dojde-li však k blokaci objektu motoru, dojde pouze k vypnutí daného objektu motoru a všech navazujících objektů, které mají blokovaný objekt v podmínce pro automatický chod.

Implementace blokadí skupiny filtrační jednotky je k vidění na obrázku 22 níže. Význam jednotlivých signálů je vysvětlený v tabulce uvedené v přílohách.

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE



Obr. 22: Blokace skupiny filtrace

## 2.2.5 Vyhodnocení funkce dvojitě výklopní klapky filtru

Dvojitá klapka filtru má za normálního provozu periodicky přecházet mezi dvěma polohami, máme tedy o její funkci k dispozici pouze dva signály, a to, že se otevřela buď horní, či spodní výklopná klapka. Pro kontrolu, zda se dvojitá výklopní klapka například nezadrhává, je nutné kontrolovat takt, ve kterém se její stavy střídají. Jelikož pro tuto funkci nebyl k dispozici již vytvořený blok, byl v rámci projektu napsán.

Blok jako výstup poskytuje pouze informaci o správné funkci jako TRUE nebo FALSE. Po detekci pokynu k chodu bloku nastává inicializační cyklus, ve kterém se nastaví paměťové bity a nedochází k vyhodnocování chyby taktu do doby, než je po tři cykly za sebou splněn požadovaný takt, po čemž končí inicializační cyklus a začíná hlídání otáček. Blok je posléze k objektu pohonu připojen přes vstup hardwarového hlídače otáček.

## 2.2.6 Regulační obvody

V průběhu realizace projektu přicházely postupně další doplňující podklady, které upřesňovaly a upravovaly požadované podmínky řízení. Mezi poslední takové úpravy patřilo i přidání specifikací pro regulaci tří měřených hodnot sušárny. Pokyny byly následující:

Ovládání regulačních obvodů má být dle pokynů možné ve dvou režimech, konkrétně v:

- Manuálním – obsluha ručně zadá požadovanou hodnotu pro řízenou veličinu.
- Automatickém – operátor zadá žádanou hodnotu pro regulovanou veličinu a regulátor vypočte potřebnou hodnotu pro řízenou veličinu. Tedy například operátor nastaví požadovanou teplotu výstupního materiálu a regulátor na základě regulačních konstant bude manipulovat s výkonem spalovací komory.

Požadované regulátory jsou následující:

- Řízení teploty materiálu na výstupu sušičky za pomoci regulace výkonu pece
- Řízení podtlaku na vstupu sušičky regulací otáček ventilátoru filtru
- Regulace teploty vzduchu na výstupu ze sušičky otevíráním klapky čerstvého vzduchu

První dva regulátory jsou v projektu implementované jako proporcionálně integrační (PI) s využitím knihovního bloku Cont\_C, vysvětleného dříve v teoretickém rozboru, a třetí pouze jako regulátor typu Bang-Bang, jelikož jde o pouhé otevírání/zavírání klapky s časovou hysterezí po dosažení stanovené teploty.

Důležité je zmínit, že stejně jako u jiných konfigurací, nelze regulátory vhodně naladit pouze pomocí simulace bez dodatečných informací, jelikož matematické modely nejsou k dispozici. Ze zmíněného důvodu je tedy konfigurace regulačních obvodů provedena pouze propojením v kódu PLC a provázáním s vizualizací. Při simulaci zmíněné později, bude pozornost věnována pouze tomu, zda se obvody zapínají ve správnou chvíli a požadují akční zásah ve správném směru. Konkrétní hodnoty regulačních konstant i dob přepínání jsou otázkou až na realizaci na místě, která je mimo rozsah této práce.

## 2.3. Implementace vizualizace

V předchozích kapitolách byla popsána jak struktura výrobní linky, tak řídicího programu. Nyní bude navázáno tvorbou vizualizace, avšak než půjde vytvořit samotné vizualizační obrazovky, je potřeba navázat spojení mezi HMI panelem a PLC. Tímto způsobem totiž lze zajistit přístup k datům uložených v PLC přímo z vizualizačních obrazovek. Dále bude rozebrán návrh metod využitelných k pozdějšímu návrhu firemního vizualizačního standardu, který ačkoliv nebyl tvořen v rámci bakalářské práce, je využit u implementace vizualizačních obrazovek. Jde o standard zabývající se zejména vizualizací a propojení dat firemně standardizovaných technologických objektů.

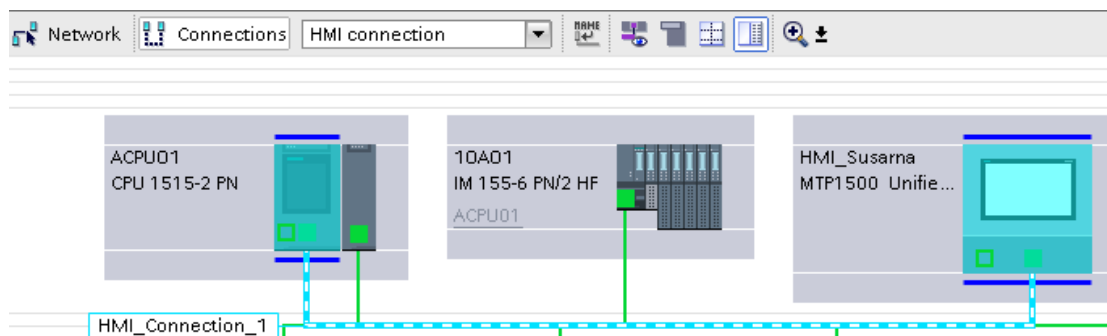
Po přípravě, popsání a začlenění firemního vizualizačního standardu do projektu přijde na řadu již samotný návrh vizualizačních obrazovek. Jelikož se jedná o poměrně malou výrobní linku, byl návrh rozložení vizualizace technologického procesu proveden na tři základní obrazovky, které budou rozebrány a ukázány později.

### 2.3.1 Propojení HMI panelu s PLC

Tvorba propojení mezi programovatelným automatem a operátorským panelem se dělí na dvě části, a to na fyzické a programové.

Pro spojení PLC a HMI zařízení je potřeba v dříve vytvořené hardwarové sestavě navázat HMI propojení a nakonfigurovat IP adresy obou zařízení tak, aby byly viditelné v rámci stejné aplikované masky podsítě. Náhled tvorby HMI spojení je na obrázku číslo 23.

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE



Obr. 23: Propojení sestavy v „hardware connections“

Fyzické propojení pak bude následně realizované s využitím technologie ProfiNet a proto je pomocí právě tohoto komunikačního média vytvořena i sběrnice síť v již dříve zmíněné hardwarové sestavě.

### 2.3.2 Přístup k datům z PLC

K datům uložených/využívaných v PLC z HMI panelu je možné přistoupit pomocí propojovacích tagů. V projektu jsou použity interní tagy na pomocné globální proměnné, například tag držící číslo datového bloku aktuálně zobrazovaného motoru. Externí tagy slouží k výměně dat s PLC. V případě externích tagů je využito jak přístup symbolický, tak absolutní, kde cílem absolutní adresace je využití multiplexovaných tagů.

V práci se využívá multiplexových tagů zejména na vlastnosti objektů zobrazených ve vyskakovacích oknech. Nehrozí tedy, že by k jednomu multiplexovanému tagu přistupovalo více vizualizačních objektů, pokud se dodrží dříve zmíněná podmínka přítomnosti pouze jednoho vyskakovacího okna v daný moment. Při multiplexování tagů se jedná primárně o adresaci na počátek struktur držících požadovaná data, například statusy motoru. Hlavní motivací k využití multiplexování je snaha o limitaci počtu využitých tagů, neb je pak možné se vejít i u větších projektů do maximálního povoleného rozsahu využitých tagů. Zároveň takto není nutné pro každý objekt vytvářet většinu tagů sloužících k vizualizaci dat a stačí dovytvářet pouze tagy potřebné k úpravě grafické stránky ikony na obrazovce.

Symbolické tagy jsou využity na signály, které nemá význam multiplexovat, nebo u kterých by docházelo k prepisování zobrazovaných dat u několika objektů zároveň. Interní tagy jsou využity například jako vnitřní proměnné reprezentující pořadové číslo daného objektu. Interní tagy jsou dále inicializovány na hodnoty shodující se s čísly vytvořených datových bloků v paměti PLC. Pomocí číslování pak lze snadno odlišit v názvu, na který objekt se zrovna operátor dívá, a i podle tohoto čísla upravovat destinaci multiplexovaného tagu.

V rámci práce je převážná většina dat čtena z instančních datových bloků, které drží veškeré informace o daném konkrétním objektu třídy definované funkčním blokem, ze kterého je tato instance vytvořena.

### 2.3.3 Návrh struktury vizualizačních objektů

Tato kapitola bude věnována popisu návrhu propojovacích a vizualizačních metod v prostředí WinCC Unified, na kterých se po předání začal vyvíjet firemní vizualizační standard pro operátorské panely SIMATIC Unified. Bude zde rozveden jak popis ať už více, či méně úspěšných navržených a otestovaných metod, tak i výsledný postup využitý v dokončeném standardu, ze kterého jsou převzaty i všechny uvedené obrázky. Je tedy vhodné zmínit, že návrh

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

a popis možných postupů je dílem autora této práce, avšak popsané dokončené metody a podoba na obrázcích již nikoliv. Výsledná podoba tedy slouží spíše pro demonstraci a vysvětlení struktury využívané dál v práci.

V rámci návrhu vizualizace výrobního procesu je třeba vytvořit grafické symboly pro potřebné technologické objekty, jako jsou objekty typu ventil, motor a další. S jejich využitím lze již poměrně snadno poskládat kostru konkrétních obrazovek. Reprezentace objektů lze obvykle vytvořit univerzálně a pak je v rámci standardu využívat i na ostatních projektech. Pro projekt sušárny vápence je však využita dříve firemně nepoužitá platforma operátorských panelů typu Unified a je tedy potřeba vizualizační objekty nejprve vytvořit. Vývoj nového firemního standardu je však nad rámec této práce, a proto byla pozornost věnována pouze návrhu prototypu, metod jeho propojení a využití funkcionalit prostředí WinCC Unified.

Vytváření vizualizačních objektů, v této pasáži demonstrováné na objektu typu motor, je nejjednodušší realizovat využitím tzv. objektů „faceplate“. Jedná se o způsob, jež vytvoří knihovni prototyp v projektu, u kterého lze navrhnout jeho vizuální stránku, nastavit mu vlastnosti a definovat, jaké vstupní parametry bude využívat. Po uložení vytvořeného faceplatu stačí již akorát jednoduše vytvářet jeho jednotlivé instance a nakonfigurovat je na tagy propojené s příslušným instančním datovým blokem obsahujícím parametry a data konkrétního motoru v programu.

Představa výsledné podoby navrženého faceplatu vychází z již stávající podoby firemního standardu vizualizačních objektů na panely SIMATIC Comfort a systémy SCADA upravený o nápady a požadavky vedoucího projektu. Výsledný soupis je takový, že na operátorské obrazovce tedy bude akorát základní vizualizace technologického objektu. Tím je myšlena v případě motoru jeho grafická stránka, ale i signalizace chodu a poruchy například změnou barvy nebo blikáním. Dalším požadovaným prvkem je znázornění, že se motor nachází v blokační podmínce a pokud má přidružený frekvenční měnič, tak na kolik procent výkonu běží. Po kliknutí na daný vizualizační objekt má dojít k otevření vyskakovacího okna, na kterém bude pomocí přepínatelných záložek možné zobrazit si detailní informace o daném objektu, jako například inicializované parametry zmíněné dříve, ale i třeba poruchy a stavy. Ve vyskakovacím okně by samozřejmě mělo být zjistitelné, o jaký konkrétní propojený objekt se jedná pomocí textového popisu.

Při návrhu byla prvně vytvořena základní ikona motoru, která měla sloužit jako token na obrazovce a dále byl hledán způsob otevírání vyskakovacího okna objektu, zobrazující jeho parametry.

Událost otevření vyskakovacího okna je třeba aktivovat pomocí java-scriptu při kliknutí na objekt. Otázkou však bylo, na jakou část objektu tuto událost nastavit. Při tvorbě byly původně navrženy a odzkoušeny níže uvedené postupy, ze kterých byla ve výsledku využita upravená verze prvního bodu:

- První způsobem řešení by bylo jednoduché nastavení události přímo na instancích faceplatu, avšak to by zavádělo nutnost individuálně konfigurovat volání vyskakovacího okna pro každou instanci. Proto bude takovýto postup ponechán jako záložní možnost.
- Druhou prozkoumávanou metodou byla aktivace skriptu při kliku do oblasti tvořené objektem, avšak ukázalo se, že konfigurovaná událost je takto nastavena pouze pro pozadí objektu, které však není překryté žádným polem.



## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

- Třetí možností bylo vytvoření neviditelného pole v nejvyšší vrstvě faceplatu. Neviditelnému poli by se následně nastavilo otevření vyskakovacího okna jako událost po kliknutí operátora, ale ukázalo se, že má-li pole atribut neviditelnosti, je ignorováno i z hlediska událostí vyvolaných operátorem (např. stisknutí, gesto atd...). Později ve vývoji se ukázalo, že požadované funkce by šlo tímto způsobem dosáhnout, pokud by se místo atributu viditelnost („visibility“) nastavil atribut neprůhlednosti („opacity“). Rozdíl mezi vlastnostmi je zřejmě takový, že viditelnost určuje, zda je objekt na obrazovce v danou chvíli aktivní, kdežto atribut opacity pouze nastaví, že je objekt průhledný, tzn., že nevykreslí jeho vizuální stránku, avšak nastavenou funkci zachová.
- U čtvrté varianty bylo počítáno s využitím komponenty „touch area“, jelikož pro panely Unified nelze využít speciálního objektu neviditelného tlačítka jako u dřívějších verzí operátorských panelů SIMATIC, i když jak bylo zjištěno, lze jeho funkce napodobit s využitím dříve zmíněného atributu opacity. V oblasti komponenty se jako události rozpoznávají pouze předdefinovaná gesta tažení a nevzniká tak reakce na samotné stisknutí. Pokud však je po jejím umístění do faceplatu zanechán prázdný skript v událostech, byla již oblast pozadí zmíněná v druhém bodě vztažena na celou instanci pokrytou dotykovou oblastí, a ne pouze na nezakryté oblasti.

Z rozepsaných postupů výše se po řadě pokusů nejvíce osvědčil trochu upravený první návrh. Ačkoliv bylo zmíněno dříve, že na platformě Unified neexistuje objekt neviditelného tlačítka, lze jeho funkce stále dosáhnout zmíněnou změnou atributu opacity. Takto přizpůsobené tlačítko se následně umístí přes instanci faceplatu a nakonfiguruje se mu skript k otevření vyskakovacího okna. Důvodů pro tuto volbu bylo hned několik, příkladem může být již jen to, že pokud bylo vyskakovací okno otevřeno z vnitřku instance faceplatu, nešlo přistoupit globálně k jeho scriptovým globálním proměnným, které by byly sdíleny pouze napříč instancemi faceplatu stejného typu. Tím by se zkomplikovalo i centrální zavírání vyskakovacích oken. Jako další důvod lze uvést to, že skripty v rámci faceplatu nabízí omezenější výběr funkcí, ve srovnání se skripty mimo něj.

Aby bylo možné využít dříve zmíněných multiplexovaných tagů, je nutné realizovat jednu ze dvou níže uvedených možností. Jinak by totiž docházelo při situaci dvou otevřených vyskakovacích oken objektu stejného typu ve stejnou chvíli k přepisování propojovacího tagu a tím pádem ke čtení stejných hodnot. Navržené metody předcházení této situaci jsou následující:

- Zajistit že bude vždy otevřeno pouze jedno vyskakovací okno objektů stejné třídy
- Zajistit, aby se hodnoty aktualizovaly pouze v časech, ve kterých bude zajištěno čtení adekvátní hodnoty.

Druhý přístup přináší komplikace v časování, a ačkoliv není nezbytně nemožný, zůstává otázkou, zda by se dala zaručit spolehlivost jeho funkce. Další nevýhodou druhé možnosti by mohla být i náročnost samotného programu, a tedy zpomalení ostatních procesů.

Možnosti s omezením počtu vyskakovacích oken nahrává i skutečnost, že předchozí verze, tj. panely řady SIMATIC Comfort panel, ani možnost více pop-up oken dle všeho neměly. Navíc, v situacích, kdy jedna vyskakovací obrazovka zabírá třeba třetinu, či čtvrtinu displeje, není toto omezení zas tak limitující a dá se tedy i případně odůvodnit.

Omezení na otevření pouze jedné vyskakovací obrazovky je tedy realizované pomocí ověření přítomnosti již zobrazovaného vyskakovacího okna všech typů objektů a případným pokusem o jejich zavření. Při otevírání okna se aktualizuje interní tagová proměnná, která určuje číslo

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

datového bloku, ze kterého budou multiplexované tagy zobrazovat data na obrazovce. Volání skriptu na otevření vyskakovacího okna a předání parametrů požadovaného objektu je znázorněno níže na obrázku 24. Na obrázku 25 je již přímo skript na otevření vyskakovacího okna motoru, spolu s korekcí jeho pozice a pokusu o zavření již otevřených vyskakovacích obrazovek.

```

1 export function _901MA01_button_OnUp(item, x, y, modifiers, trigger) {
2
3 try {
4     GM.Motor_PopUp(HMIRuntime.Tags.SysFct.GetTagValue("DB_motor_001"),x,y);
5 }
6 catch (err) {
7     let tag1 = Tags("Error_text");
8     tag1.Write(err);
9 }
10 }
    
```

Obr. 24: Volání skriptu na otevření vyskakovacího okna z neviditelného tlačítka překrývajícího instanci faceplatu objektu motoru

```

1 export function Motor_PopUp(DB_Tag_Number, X, Y) {
2     Tags("DB_Number").Write(DB_Tag_Number);
3     let x_position = X;
4     let y_position = Y;
5     let x_screen = Tags("X_Screen").Read() - Tags("X_Window_Motor").Read() - Tags("X_correction").Read();
6     let y_screen = Tags("Y_Screen").Read() - Tags("Y_Window_Motor").Read() - Tags("Y_correction").Read();
7
8     //Position correction
9     if (x_position > x_screen)
10    {
11        x_position = x_screen;
12    }
13    if (y_position > y_screen)
14    {
15        y_position = y_screen;
16    }
17    if (x_position < 0)
18    {
19        x_position = 0;
20    }
21    if (y_position < 0)
22    {
23        y_position = 0;
24    }
25
26    //Try closing open pop-ups
27    try
28    {
29        HMIRuntime.UI.SysFct.ClosePopup("/PopUp_Motor");
30    }
31    catch
32    {}
33
34    •
35    •
36    •
99    try
100   {
101       HMIRuntime.UI.SysFct.ClosePopup("/PopUp_Encoder");
102   }
103   catch
104   {}
105   HMIRuntime.UI.SysFct.OpenScreenInPopup("PopUp_Motor", "PopUp_Motor", false, "", x_position, y_position, false, undefined);
106 }
    
```

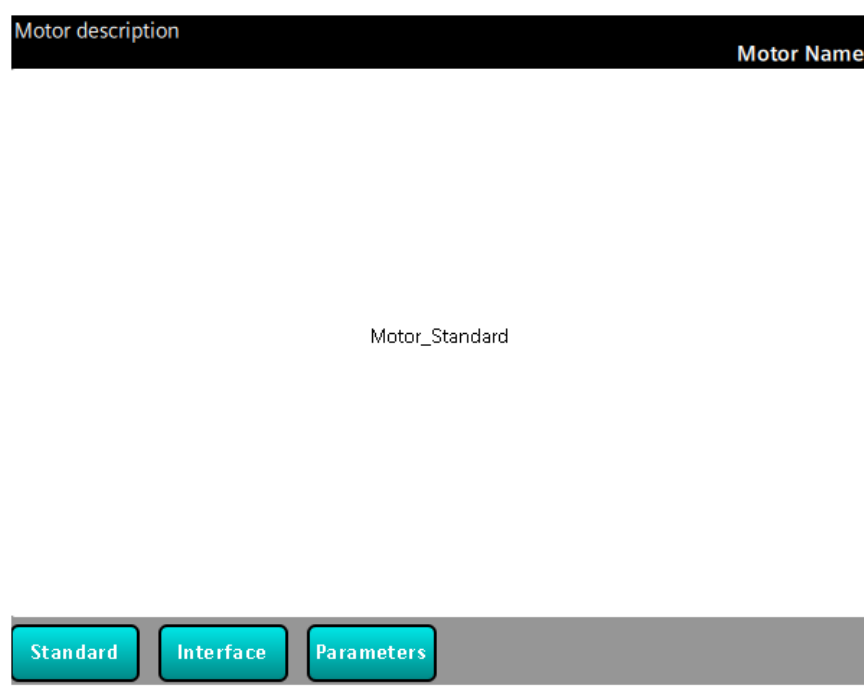
Obr. 25: Korekce pozice, pokus o zavření otevřených vyskakovacích oken a následné otevření požadovaného vyskakovacího okna

Vyskakovací obrazovka má být dle požadavku členěná na jednotlivé záložky, které seskupují atributy objektu do skupin. Při návrhu byly prozkoumány tři následující pokusy při hledání možnosti dovolující implementovat požadovanou funkci záložek, konkrétně:

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

- První a potenciálně nejméně pracnou variantou, by bylo vytvoření obsahu jednotlivých záložek jako samostatného faceplatu a následně vytvoření jejich instance ve faceplatu okna, případně jejich seskupení do jedné grupy, které by se dal plošně nastavit atribut viditelnosti, avšak tato metodika není podporovaná ze strany vývojového prostředí v případě panelů Unified.
- Druhou prozkoumávanou možností bylo rozdělení objektů do jednotlivých vrstev faceplatu, kde prostředí toto členění podporuje na 32 vrstev, do kterých se dají ukládat jednotlivé objekty. Členění by probíhalo podle příslušnosti k dané záložce a následně by se nastavil jednotně status viditelnosti pro celou vrstvu. Realita je taková, že jednotlivým vrstvám skutečně atribut viditelnosti modifikovat lze, avšak zdá se, že pouze ve vývojovém náhledu, a nikoliv za běhu v reakci na splnění jisté podmínky.
- Třetí, nejméně elegantní, za to funkční variantou je individuální nastavení viditelnosti každému objektu obsaženému v dané záložce.

Ve výsledku využitá varianta se nejvíce podobá prvnímu bodu, kdy místo záložek tvořených jako samostatný faceplate jsou vytvořeny jako samostatné obrazovky. Takto vytvořené obrazovky lze totiž posléze promítat přes objekt screen window přímo do vybrané části vyskakovacího okna. Výsledná struktura pop-up okna je k vidění na obrázku 26 níže, kde názvy a popisy zobrazovaného objektu jsou svázány s textovým listem a bílá část středu okna značí právě využitou komponentu screen window.



Obr. 26: Náhled struktury vyskakovacího okna

V další fázi dochází již k provázání potřebných tagů s tlačítky a dalšími případnými objekty a následně propojení vstupních textových listů se zobrazovaným popisem. Zde je třeba vzít v potaz, že při čtení z multiplexovaného tagu, který odkazuje například na strukturu nějakého datového bloku, jsou v neoptimalizovaných blocích data ukládána ve formátu big endian, a je tedy potřeba při zobrazování hodnot individuálních bitů počítat se změnou jejich pořadí vůči formátu v datovém bloku.

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

Navržený prototyp propojení vizualizace standardizovaného objektu byl posléze předán kolegovi, který na základě přijatého návrhu pokračoval a dokončil jejich vývoj. Navrhl tedy jak vizuální stránku, která je vidět na obrázku x výše, tak i některé dodatečné úpravy ohledně vyskakovacích obrazovek a realizoval celkovou funkci. Důležité je tedy zmínit že se tato kapitola, jak již bylo zmíněno na začátku, netýká vývoje standardizovaných objektů, avšak pouze měla za úkol prozkoumat a navrhnout metody k dosažení požadovaného výsledku, na jejichž základě byl započat samotný vývoj kolegou.

### 2.3.4 Tagové listy pro vizualizaci

Vizualizace technologických objektů bude na obrazovkách tvořena již na základě dokončeného standardu vycházejícího z předaného návrhu rozepsaného v předchozí kapitole. Aby šel standard zdárně zakomponovat do projektu, je třeba správně navázat i propojovací tagy na instance faceplatů, vyskakovací okna a později do projektu zahrnou i grafické listy.

Struktura tagových listů je tedy taková, kde pro vyskakovací okno existuje pro všechny instance objektů daného typu centrální sdílený seznam multiplexovaných tagů, u kterých je offset adres posouván využitím pomocné proměnné obsahující číslo instančního datového bloku momentálně zobrazovaného objektu a jehož strukturu lze vidět na obrázku 27. Ve zmíněném seznamu tagů jsou například tagy odkazující na strukturu datového bloku obsahující jeho stavy, nebo i například parametry definované v inicializační funkci. Jakmile tedy bylo jednou vytvořeno provázání tagů s vyskakovacím oknem v rámci tvorby standardu, stačí tento centrální list jen vytvořit v projektu, a při volání vyskakovacího okna z události tlačítka pouze nastavit číslo požadovaného datového bloku. Teoreticky by šlo zmíněným postupem provázat libovolné množství objektů bez zásahu do tagových listů, avšak z důvodů vysvětlených v následujícím odstavci je číslo datového bloku, které se předává funkci volající vyskakovací okno, definováno jako tag s pevně nastavenou interní hodnotou.

S_Mux_Motor						
Name	Data type	Connection	PLC name	PLC tag	Address	Access mode
Motor_Actual_SP	Real	HMI_Conne...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD16	<absolute access>
Motor_ChangeDirDelay	Time	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD182	<absolute access>
Motor_Command_DWord	DWord	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD60	<absolute access>
Motor_Counter_Hours	DInt	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD116	<absolute access>
Motor_Counter_Starts	DInt	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD122	<absolute access>
Motor_Counter_Time_Stamp	Date_And_Time	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBX126.0	<absolute access>
Motor_FeedbackDelay	Time	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD166	<absolute access>
Motor_FeedbackStopDelay	Time	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD170	<absolute access>
Motor_FlowDelay	Time	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD194	<absolute access>
Motor_Horn_Startup_Time	Time	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD198	<absolute access>
Motor_Man_SP	Real	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD64	<absolute access>
Motor_Max_Value	Real	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD148	<absolute access>
Motor_Min_Value	Real	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD144	<absolute access>
Motor_Object_Definition_DWord	DWord	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD136	<absolute access>
Motor_Object_Status_DWord	DWord	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD76	<absolute access>
Motor_OFFDelay	Time	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD178	<absolute access>
Motor_ONDelay	Time	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD174	<absolute access>
Motor_Signal_Status_DWord	DWord	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD80	<absolute access>
Motor_Signals_Definition_DWo...	DWord	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD140	<absolute access>
Motor_SpeedMonDelay	Time	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD186	<absolute access>
Motor_SpeedMonTact	Time	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD190	<absolute access>
Motor_Unit_Code	Int	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBW134	<absolute access>
Motor_VDF_Speed	Real	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD84	<absolute access>
Motor_Waiting_Startup_Time	Time	HMI_Connectio...	ACPU01	->Multiplex tag->	%DB[DB_Number].DBD202	<absolute access>

Obr. 27: Přehled struktury listu multiplexovaných tagů pro provázání proměnných z PLC s vizualizací na vyskakovacích oknech motoru

Pro samotný faceplate již nelze využít dynamické multiplexované tagy z důvodu vzájemného přepisování pomocné adresní proměnné, jak již bylo zmíněno v kapitole zabývající se návrhem struktury vizualizačních objektů. Je tedy nutné pro každou unikátní instanci vytvořit i náležitý set tagových proměnných, které jsou posléze využity ke grafické vizualizaci momentálního stavu technologického objektu. Jejich interní využití je tedy zejména na konfiguraci dynamické

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

viditelnosti například symbolů značících blokaci motoru, či na stavové přepínání zobrazované ikony čtené z grafických listů, o čemž bude ještě zmínka později.

Struktura setu tagů pro instance faceplatu je tvořena listy naplněnými tagy odkazujícími se na stejnou adresu, avšak odlišného datového bloku. Pro odlišení je v názvu tagu koncovka shodná s posledními třemi ciframi instančních bloků pro daný technologický objekt. V projektu je totiž pro každý typ standardizovaných objektů daný přesný rozsah po tisíci položkách, ze kterého by mělo vycházet číslování instančních datových bloků objektu daného typu. Na obrázku 28 níže, ukazující list tagů navázaných na strukturu definic motorů, odkazuje například položka DB\_motor\_001 v adresním sloupci na datový blok s číslem 10001 a toto číslo má pevně nastavené přes interní proměnnou definovanou v jiném tagovém listu. Přes tuto proměnnou dochází zároveň k propsání čísla požadovaného datového bloku do vyskakovacího okna, jak bylo zmíněno dříve, a k propsání požadovaného popisu a názvu technologického objektu z textového listu, jehož tvorba bude vysvětlena níže.

S0_Motor_Definition						
Name	Data type	Connection	PLC name	PLC tag	Address	Access mode
S0_MOTOR_DEFINITION_001	DWord	HMI_Conne...	ACPU01	<Multiplex tag>	%DB[DB_motor_001].DBD136	<absolute access>
S0_MOTOR_DEFINITION_002	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_002].DBD136	<absolute access>
S0_MOTOR_DEFINITION_003	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_003].DBD136	<absolute access>
S0_MOTOR_DEFINITION_004	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_004].DBD136	<absolute access>
S0_MOTOR_DEFINITION_005	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_005].DBD136	<absolute access>
S0_MOTOR_DEFINITION_006	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_006].DBD136	<absolute access>
S0_MOTOR_DEFINITION_007	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_007].DBD136	<absolute access>
S0_MOTOR_DEFINITION_008	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_008].DBD136	<absolute access>
S0_MOTOR_DEFINITION_009	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_009].DBD136	<absolute access>
S0_MOTOR_DEFINITION_010	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_010].DBD136	<absolute access>
S0_MOTOR_DEFINITION_011	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_011].DBD136	<absolute access>
S0_MOTOR_DEFINITION_012	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_012].DBD136	<absolute access>
S0_MOTOR_DEFINITION_013	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_013].DBD136	<absolute access>
S0_MOTOR_DEFINITION_014	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_014].DBD136	<absolute access>
S0_MOTOR_DEFINITION_015	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_015].DBD136	<absolute access>
S0_MOTOR_DEFINITION_016	DWord	HMI_Connectio...	ACPU01	<Multiplex tag>	%DB[DB_motor_016].DBD136	<absolute access>

Obr. 28: Přehled struktury listu propojovacích tagů pro provázání proměnných z PLC s grafickou vizualizací faceplatu motoru

Podobným způsobem jako na obrázku výše jsou definované i zbylé potřebné proměnné pro všechny typy faceplatu firemně standardizovaných vizualizačních objektů, kde pak při jejich navázání na instance faceplatu stačí měnit akorát číselné koncovky v názvech požadovaných tagů pomocí funkce najít a nahradit.

### 2.3.5 Tvorba textových listů

Dále je vhodné věnovat pozornost i informacím zobrazovaným ve vyskakovacích oknech vizualizačních objektů. Signály z PLC obsahující statusy a definice daného objektu jsou již sice provázané, ale je třeba zajistit operátorovi i jednoznačnou rozlišitelnost objektu podle jeho názvu a popisu. Za tímto účelem jsou použity textové listy, které propíší název i popis objektu ve vizualizaci ve všech jazycích projektu, bude-li do těchto jazyků přepnut. Textovému poli sice mnoho jazyčný text nastavit lze, ale jelikož je textové pole na vyskakovací obrazovce sdílené pro všechny instance objektů, které dané okno mohou vyvolat, je nutné zajistit i odlišení zobrazovaného textu v závislosti na objektu, který je právě zobrazován. Je proto nutné využít dynamické konfigurace textového pole pomocí textového listu a modifikovat zobrazovaný text v závislosti na číslu datového bloku vizualizovaného objektu.

Tvorba textového listu spočívá ve vytvoření jednotlivých listů pro popis i název každého využitého technologického objektu. Stejně jak tomu bylo při tvorbě tag listu, jsou i zde indexy položek textového listu voleny tak, aby se indexy textu shodovaly s číslem datového bloku korespondujícího objektu. Pro jednodušší editaci textu ve zbylých jazycích projektu a využití již vytvořených popisů (například networků) v programu PLC, lze využít funkci exportu

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

textových listů do formátu excelu. Pomocí nástroje filtrů a funkcí excelu lze snadno modifikovat již existující text, či přidávat nový. Velkou výhodou exportovaného listu je i to, že je-li potřeba doplnit texty pro další jazyky projektu, stačí izolovat již existující textový popis do nového excelovského souboru a s využitím online nástrojů celý soubor přeložit do požadovaného jazyka. Takto přeložený text lze opět nahrát do příslušného sloupce původního listu, zkontrolovat pouze přítomnost chyb a list opět naimportovat do projektu. Náhled na strukturu textového listu je k dispozici na dříve uvedeném obrázku číslo 13. Překlady textových listů jsou provedeny pro účely projektu zatím pouze strojově. Od zákazníka byly textové popisy obdrženy v ruštině a nepříliš dobré češtině, které byly následně přeloženy do angličtiny.

### 2.3.6 Statické vizualizační ikony

Na vizualizačních obrazovkách je vhodné mít nejen pouze měřené, či řízené veličiny, ale i grafickou reprezentaci technologických částí, kterých se řídicí proces týká. V případě této práce se jedná o znázornění jak rotačního bubnu, filtru tak i spalovací komory, jelikož jde o komponenty řízené, či alespoň velice významné pro samotný výrobní proces.

Ikony lze sice vytvořit ručně kombinací základních elementů, avšak výsledný produkt se hodí spíš pro dočasnou reprezentaci na obrazovkách. Problém spočívá v tom, že vytvořený objekt nepůsobí příliš prostorovým dojmem. Lze sice namítnout, že se dá onen dojem dotvořit, avšak to by stálo spoustu času a trochu stylistického nadání. Většinou je jednodušší zkusit najít obrázek reprezentující požadovaný objekt na internetu a po sléze ho importovat jako ikonu grafického objektu, nebo si ho sestavit z již existujících grafických objektů.

Sestavení ikony z mixu grafických objektů doprovází problém změny měřítka výsledné tvorby. Jak již bylo zmíněno dříve u tvorby návrhu vizualizačního objektu motoru, u využívaného operátorského panelu Unified nelze na rozdíl od předchozích verzí panelů Comfort a Basic využít grupování objektů do jednoho celku. Aby se tedy šlo vyhnout změně měřítka všech individuálních částí, je třeba prozkoumat více možností. Níže jsou v bodech rozvinuty postupy vedoucí k požadovanému výsledku, spolu s nevýhodami s nimi spojenými:

- Jednou možností je umístit vytvořený objekt na samostatnou obrazovku a oříznout velikost okna na jeho rozměry. Následně lze na místo, kde se má cílená grafická reprezentace nacházet vložit screen window a odkázat na obrazovku, v níž se nachází vytvořený objekt. Nastaví-li se totiž držiči obrazovky atribut škálování do velikosti okna, přizpůsobí se velikostně i vytvořený grafický objekt. Avšak velkou nevýhodou tohoto přístupu je skutečnost, že na tvořené vizualizaci není vidět propisovaná obrazovka do objektu screen window do doby, než je projekt buď simulován, či nahrán do fyzického panelu. Jinými slovy, přesné pozicování propsané vizualizační ikony by bylo nutné dělat metodou pokus omyl a po každé úpravě ověřovat v simulátoru.
- Druhým přístupem je tvorba jednoho grafického objektu z ikony vytvořenou kombinací několika grafických tokenů z globální knihovny. Ikonu by takto šlo vytvořit buď v nějakém grafickém editoru, či v momentě, ve kterém je již vytvořená kombinace na vizualizační obrazovce, přenosem do formátu obrázku například pomocí nástroje snipping tool od Windows. Obrázky pak totiž jdou již vkládat do grafických objektů, ať už v projektové knihovně, nebo v grafických listech. Pořízením obrázku však práce nekončí, protože obsahuje i pozadí, které by překrylo okolí vytvořené ikony. Pozadí jde potlačit propsáním průhledné vrstvy na jeho barvu, ideálně dopředu změněnou na bílou, například v nástroji GIMP. Mezi nevýhody zmíněného postupu lze uvést například to, že po vytvoření již nelze bez opakování celého postupu ikonu modifikovat a to, že pořízený

## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

obrázek zmíněným postupem nemusí být nejvyšší kvality<sup>5</sup>. Tedy při změně velikosti by mohla výsledná grafická ikona působit rozmazaně.

- Poslední prozkoumanou a v projektu i aplikovanou variantou bylo spojení shluku grafických objektů do jednoho faceplatu. Postup spočívá pouze v označení všech potřebných objektů na obrazovce a kliknutí na možnost vytvořit faceplate. Nevýhodou zmíněného přístupu je však to, že do objektu faceplate se automaticky nepropíší grafické objekty z globální knihovny. Respektive propíší, avšak bez jejich vizuální stránky, tu je následně nutné opět doplnit. Pro doplnění grafického náhledu byly vytvořeny grafické objekty v knihovně projektu, do kterých byly vloženy ikony z globální grafické knihovny. Ikony z takto vytvořené projektové knihovny jdou posléze vložit pomocí „drag and drop“ do požadovaného držiče grafiky v dříve vytvořeném faceplatu.

Vytvořené statické ikony jsou k vidění na již dříve uvedeném obrázku 16, kde ikony zároveň slouží pro vysvětlení struktury linky a k popsání dostupných technologických objektů pro účely řízení.

### 2.3.7 Dynamické vizualizační ikony

Pro instance faceplatů na vizualizačních obrazovkách je rovněž jako v předchozí kapitole potřeba vytvořit dodatečné grafické ikony. Využitím grafických listů určených dle instance faceplatu lze totiž zabránit situaci, kdy by pro každý typ technologického objektu musel být vytvořen samostatný faceplate s odlišnou vizuální stránkou, ikdyž by jeho funkce zůstala stejná. Při využití grafických listů totiž stačí, když je list zaslán do instance faceplatu a z jejího vnitřku je následně zobrazena grafická reprezentace z komponenty „symbolic I/O field“. Konkrétní grafika je z listu vybrána na základě aktuální hodnoty propojovacího tagu.

Jednotlivé grafické ikony byly tedy vytvořeny postupem uvedeným v druhém bodě předchozí sekce. Tentokrát se však, i díky jednoduchosti a dopředu známé podobě, jednalo o tvorbu v grafickém editoru a následné vložení do projektu po provedení úpravy pro průhlednost pozadí.

Každá grafická ikona byla navržena v základní verzi ve třech barvách. Konkrétně:

- Šedé pro vizualizaci zastaveného stavu, či neutrálního stavu digitálního objektu
- Zelené pro znázornění aktivního stavu
- Červené k signalizaci poruchy, nebo dosažení kritických hladin

Ikony ve své neaktivní podobě jsou k nahlédnutí níže na obrázku 29. První dvě ikony byly převzaty již s firemním vizualizačním standardem a symbolizují jak detailně, tak jednoduše objekt pohonu. Další uvedené ikony již byly do jisté míry vytvořeny dle předlohy pro účely tohoto projektu a jedná se popořadě o: Jednotku ohřevu, řídicí jednotku, brzdu pohonu, ventilátor, detekci vysokého a nízkého tlaku, a konečně vysoká či nízká detekovaná hladina materiálu v nádrži. Z hlediska programového standardu je prvních šest ikon reprezentací objektů motoru, kdežto poslední čtyři ikony vizualizují objekty digitálů.



<sup>5</sup> V případě že ikona nebyla vytvořena v grafickém editoru, ale byla pořízená fotkou obrazovky a došlo následně ke změně rozlišení




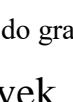
VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

Obr. 29: Náhled využitých dynamických vizualizačních ikon projektu (vysvětlivky výše)

Takto vytvořené grafické ikony vizualizačních objektů je třeba následně vložit do grafických listů, ze kterých bude posléze propána konkrétní barva ikony na vizuální stránky instance facepletu dle stavu příslušného technologického objektu. Struktura textového listu je uvedena na obrázku 30 níže.

Graphic lists			
Name	Selection	Comment	
Fan_simple	Value/Range		
Heating_Simple_Circle	Value/Range		
Heating_Simple_Square	Value/Range		
MachineState	Value/Range		
Motor_A_L	Value/Range		
Motor_A_R	Value/Range		
Motor_Brake	Value/Range		
Motor_simple	Value/Range		
Piston_gray_IN	Value/Range		
Piston_lock_IN	Value/Range		

Graphic list entries				
Default	Value	Graphic na...	Graphic	
<input type="radio"/>	0	Brake_Gray		
<input type="radio"/>	1	Brake_Green		
<input type="radio"/>	2	Brake_RED		
<input checked="" type="radio"/>	Default entry	Brake_Gray		

Obr. 30: Vložení vytvořených ikon do grafického listu

### 2.3.8 Uživatelské rozhraní obrazovek

Jako další potřebný aspekt vizualizace je nutné vytvořit strukturu navigačního panelu mezi obrazovkami a případné dělení na záhlaví a zápatí obrazovek. Navigační panel byl stanoven jako součást zápatí spolu s možností prokliku do obrazovky nastavení. Záhlaví následně bude zobrazovat informace o přihlášeném operátoru, datumu s časem, momentálně nakonfigurovaný jazyk projektu a možnost zobrazení přehledu alarmů. Jak záhlaví, tak zápatí je vloženo na každou obrazovku individuálně pomocí komponenty screen window. Šlo by sice vytvořit jednu centrální obrazovku, na které by bylo zobrazeno záhlaví se zápatím a pak pouze promítat do jejího středu požadovaný obsah<sup>6</sup>, avšak tento způsob nepůsobil příliš slibně. Jako důvod lze uvést například to, že nelze jednoduše modifikovat obsah navigačního panelu v závislosti na obrazovkách, což by mohlo být výhodné, pokud by bylo třeba zobrazit větší množství obrazovek. Dalším důvodem je skutečnost, že při promítání obrazovek na obrazovku centrální docházelo k „uříznutí“ části navrženého okna, a musela se tedy tato skutečnost brát při návrhu v potaz.

Navigační panel jako takový obsahuje tlačítko pro načtení domovské obrazovky, možnost návratu na předchozí obrazovku a proklik na konkrétní implementované obrazovky. Při přesunu na novou obrazovku je název staré obrazovky uložen do zásobníku, ze kterého jsou posléze podle principu poslední dovnitř, první ven vyčítány názvy obrazovek pro realizaci funkce návratu na předchozí obrazovku. Další přidanou funkcionalitou je zešedivění a odepření možnosti stisknout tlačítko odkazující na právě otevřenou obrazovku. Jde o funkci realizovanou poměrně triviálně, a

<sup>6</sup> Stejně tak jak funguje navigace v šabloně od Siemensu pro platformu Unified



## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

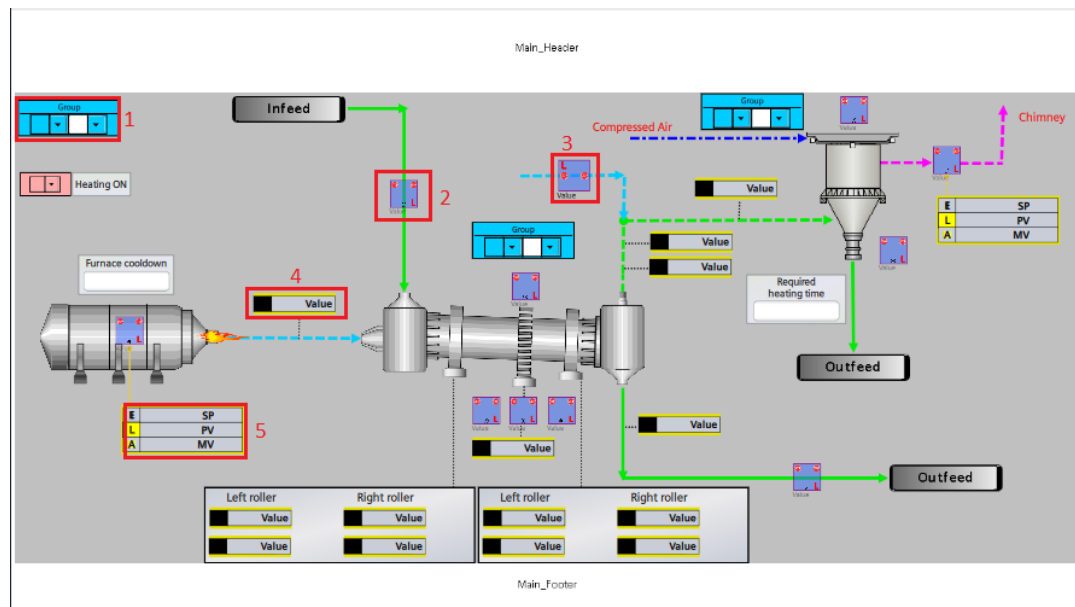
to pomocí přepsání pomocné proměnné při otevření obrazovky interním indexem, který jí byl přiřazen. Na základě zmíněné pomocné proměnné je modifikován atribut `op_control` u navigačních tlačítek, čímž se zároveň zařídí funkce výše popsaná. Atribut totiž specifikuje, zda operátor může objekt používat.

### 2.3.9 Kompletace vizualizačních obrazovek

Jak již bylo rozebráno v úvodu kapitoly implementace vizualizace, byl návrh rozložení vizualizace technologického procesu rozčleněn na tři základní vizualizační obrazovky. První přehledovou obsahující důležité informace o celém výrobním procesu a jeho částí. Druhá obrazovka obsahuje detail funkce bubnu rotační sušárny a třetí zobrazuje detail filtrační části. Navrch k těmto třem obrazovkám je ještě dodatečná vizualizace zaměřená na vizualizaci stavu sepnutí jističů a tlačítek nouzového zastavení.

Jelikož jsou nyní hotovy přípravy k tvorbě obrazovek, stačí na ně již jen umístit a propojit všechny potřebné objekty. Idea je taková, že na přehledové obrazovce se budou nacházet pouze klíčové informace o celém výrobním procesu, kdežto na detailních obrazovkách budou i informace rozšiřující týkající se dané technologie a informace o stavu předcházejícího či navazujícího procesu. Na níže uvedených obrázcích je tedy k vidění struktura navržených obrazovek.

Obrazovka číslo 31 konkrétně ukazuje přehledovou obrazovku během vývoje. Jelikož program ještě neběží, nejsou do instancí `faceplatu` propsané dynamické ikony z grafických listů a nedochází ani k zobrazení obrazovek navigačních panelů komponentou `screen container`. Barvy cest ať už materiálu, či vzduchu nejsou definitivní a vycházejí provizorně z technologického schématu. Legenda čistě pro účely zprávy je na obrázku 32.



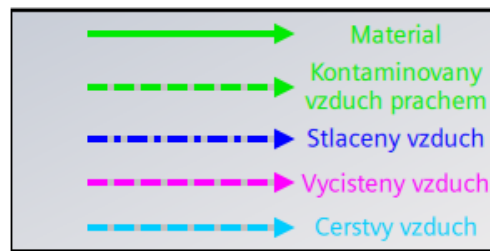
Obr. 31: Vývojový náhled přehledové obrazovky sušárny vápence

Vysvětlivky pro popisky na obrázku 31 výše jsou následující:

1. Instance `faceplatu` objektu skupiny
2. Instance `faceplatu` objektu motoru
3. Instance `faceplatu` objektu klapky
4. Instance `faceplatu` objektu analogu

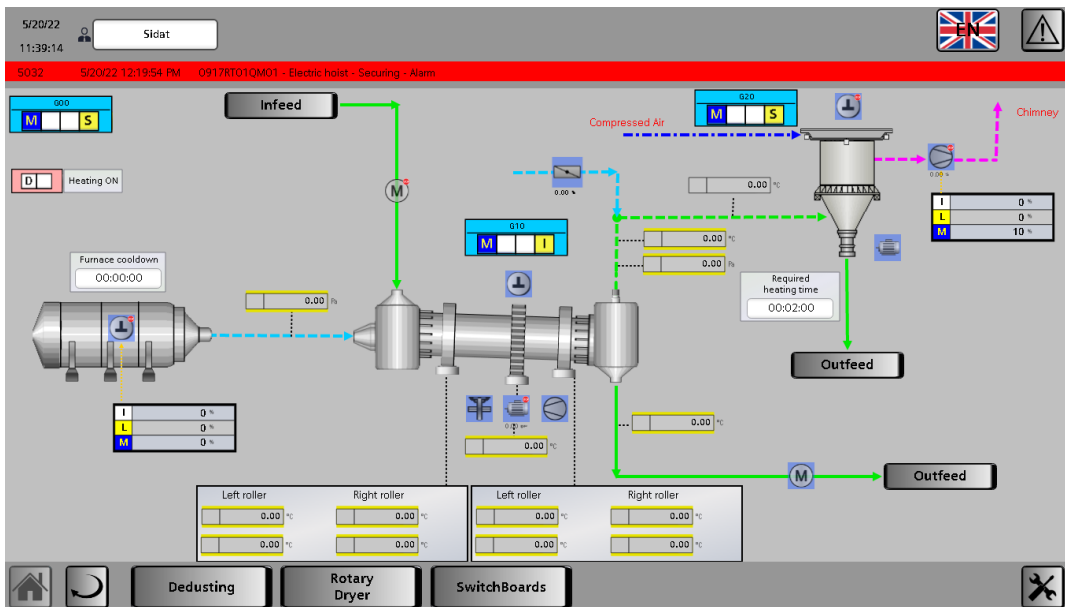
VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

5. Instance faceplatu objektu regulátoru



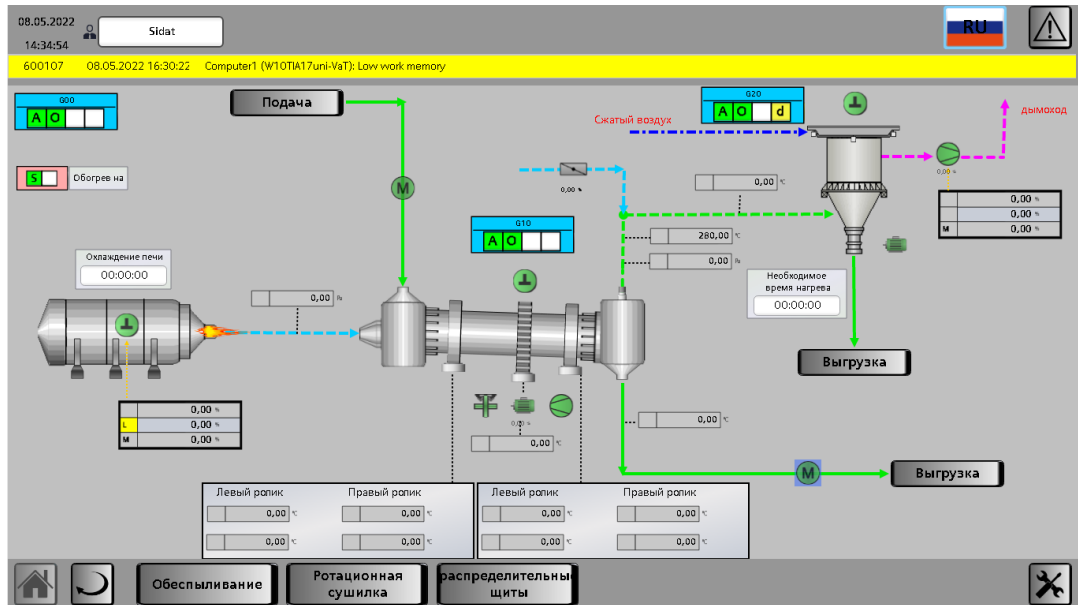
Obr. 32: Legenda významů spojovacích čar pro účely zprávy (z projektu odstraněno)

Na obrázcích 33 a 34 uvedených níže je prvně vidět přehledová obrazovka zastavené (značené šedivou barvou grafických ikon) linky v manuálním režimu (značeno modrým podbarvením vizualizačních objektů). Druhý obrázek znázorňuje linku sušení vápence již v plném provozu (znázorněno zelenými vizualizačními ikonami) s přepnutým jazykem projektu do ruštiny.

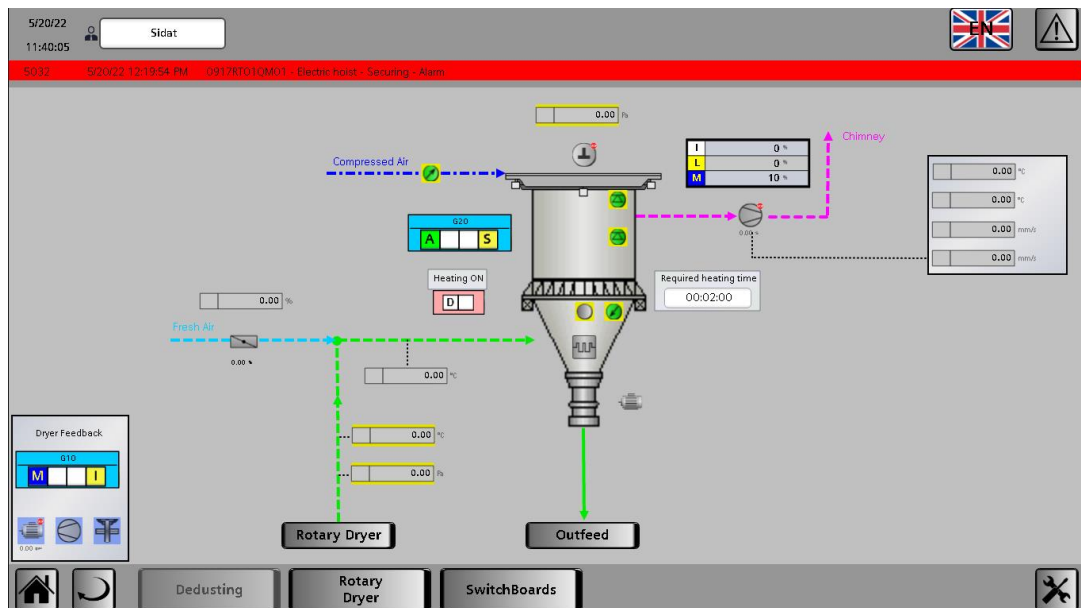


Obr. 33: Náhled navržené přehledové obrazovky v zastaveném manuálním režimu linky

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

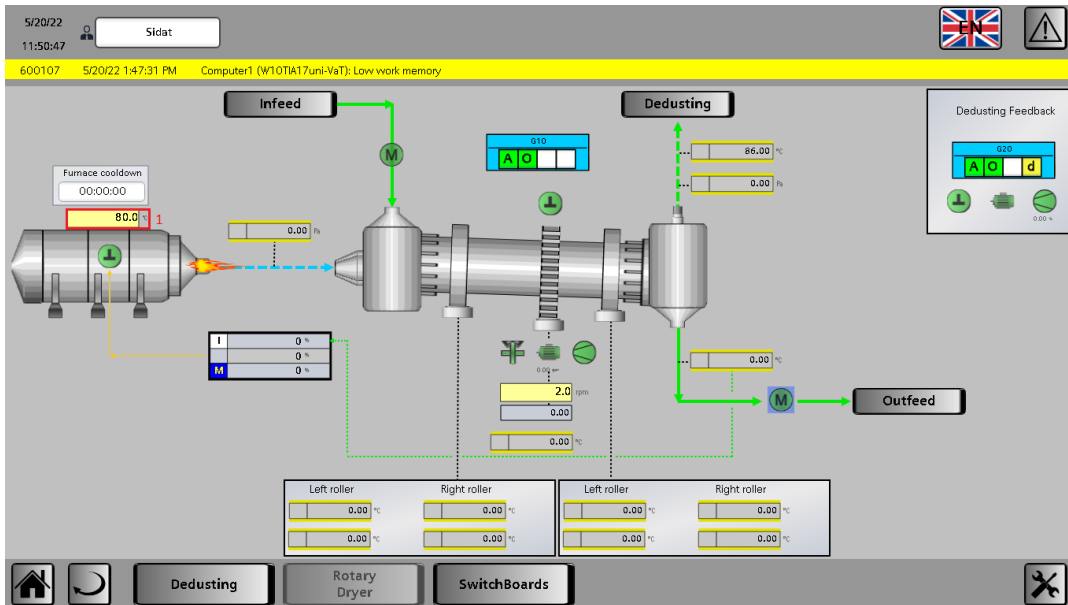


Obr. 34: Přehledová obrazovka linky za chodu s přepnutým projektovým jazykem do ruštiny  
Na níže uvedených dvou obrázcích 35 a 36 jsou znázorněny postupně obrazovky s detailem zastavené filtrační jednotky a posléze obrazovka detailu rotační sušárny za provozu linky.



Obr. 35: Náhled navržené obrazovky filtru během vypnuté linky

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE



Obr. 36: Náhled navržené obrazovky rotační sušárny za simulovaného chodu linky

Vysvětlivky pro popisky na obrázku 36 výše jsou následující:

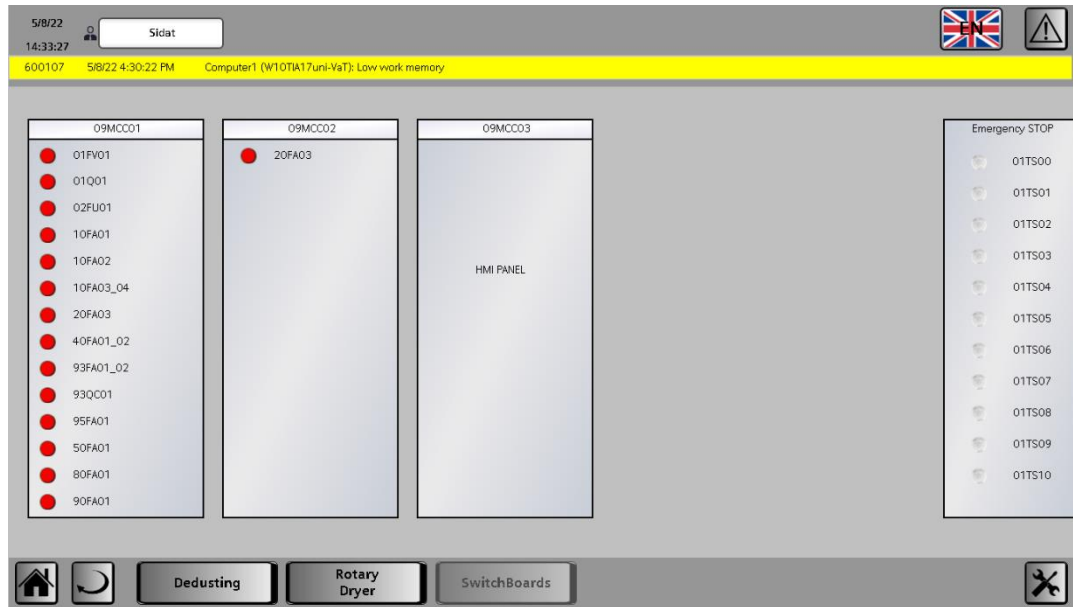
1. Pole objektu setpoint. Umožňuje operátorovi specifikovat hodnotu, na kterou se má například vyhřát spalovací komora, nebo na kolik otáček za minutu se má roztočit pohon bubnu sušárny.

Na obrázku 37 uvedeném níže lze vidět obrazovku alarmů ve chvíli, kdy objekty nejsou zjednodušeny pro simulační účely a vzniknou na nich chyby vlivem neaktivních vstupních signálů. Obrázek 38, rovněž níže, zobrazuje navrženou operátorskou obrazovku obsahující stavy jističů a nouzových tlačítek linky.

ID	Alarm class	Alarm text	Raise time	Status text
4	S_Alarm	Motor 0901EM01 - Drum dryer - brake - drive is out of order	5/8/22 4:25:4	Incoming
5	S_Alarm	Motor 0901EM01 - Drum dryer - brake - error from VFD	5/8/22 4:25:4	Incoming
6	S_Alarm	Motor 0904MA01 - Filter fan - drive is not ready	5/8/22 4:25:4	Incoming
7	S_Alarm	Motor 0903MA01 - Glove filter - double flap - communication fault with VFD	5/8/22 4:25:4	Incoming
8	S_Alarm	Motor 0903MA01 - Glove filter - double flap - communication fault with VFD	5/8/22 4:25:4	Incoming
9	S_Alarm	Motor 0901EM01 - Drum dryer - brake - drive not stopped	5/8/22 4:25:4	Incoming
10	S_Alarm	Motor 0904MA01 - Filter fan - drive is not ready	5/8/22 4:25:4	Incoming
11	S_Alarm	Motor 0901EM01 - Drum dryer - brake - error from VFD	5/8/22 4:25:4	Incoming
12	S_Alarm	Motor 0901EM01 - Drum dryer - brake - drive is out of order	5/8/22 4:25:4	Incoming
13	S_Alarm	Motor 0904MA01 - Filter fan - drive is not ready	5/8/22 4:25:4	Incoming
14	S_Alarm	Motor 0901EM01 - Drum dryer - brake - drive is out of order	5/8/22 4:25:4	Incoming
15	S_Alarm	Motor 0901EM01 - Drum dryer - brake - error from VFD	5/8/22 4:25:4	Incoming
16	S_Alarm	Motor 0901EM01 - Drum dryer - brake - drive not stopped	5/8/22 4:25:4	Incoming
17	S_Alarm	Motor 0903MA01 - Glove filter - double flap - communication fault with VFD	5/8/22 4:25:4	Incoming
18	600108	SystemAlarm - Computer1 (W10TIA17uni-VaT): Very low work memory	5/8/22 4:26:2	Incoming/Outgoing
19	600107	SystemWarn - Computer1 (W10TIA17uni-VaT): Low work memory	5/8/22 4:30:2	Incoming

Obr. 37: Přehled struktury vizualizace alarmového hlášení

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE



Obr. 38: Náhled obrazovky přehledu rozvaděčů a tlačítek nouzového zastavené

Jelikož je již nyní vytvořená vazba mezi daty programu PLC a vizualizovanými daty a jsou vytvořené obrazovky vizualizace výrobního procesu, stačí již jen přikročit k simulaci a projekt následně otestovat. Právě testování s využitím simulátorů a dodatečný popis vizualizačních obrazovek je rozebrán v následující kapitole.

### 3. Simulace výsledků

Simulací ve vizualizačním prostředí jde, jak ověřit funkci navržených obrazovek, tak i daleko jednodušeji otestovat vytvořené programové vazby. Z toho důvodu se s ověřením programových podmínek čekalo na dokončení první verze vizualizace. Kontrola vytvořené vizualizace a programových vazeb byla provedena pomocí kombinace simulátorů kódu PLC Sim a simulátoru vizualizačních obrazovek WinCC Unified RT. Důležité je brát v úvahu, že pomocí simulace nelze ani zdaleka ověřit kompletní funkci programu, natož odladit regulátory bez dodatečných informací. Důvodů je několik, ale mezi nejpodstatnější pochopitelně patří absence vstupních signálů, které je díky tomu nutné aspoň z části simulovat. Jak již bylo vysvětleno u sekce zabývající se simulátorem PLC Sim, lze u něj totiž po rozšíření náhledu naimportovat tagy z projektu a simulovat vstupní periferie, které by běžně přes nástroj watch table<sup>7</sup> nastavit nešly<sup>8</sup>.

Následující sekce se budou zabývat jak popisem zprovoznění simulátoru obrazovek, tak popisem struktury a výsledků dosažených simulací. Pro ověření funkčnosti programu a vizualizace byla navržena následující struktura testování:

- První cílem testů je kontrola provázaných signálů, tedy zda objekt vizualizuje správné informace a nedošlo například k navázání špatného propojovacího tagu.
- Druhým bodem zájmu je ověření navázání technologických objektů na objekt skupiny. Testování se tedy zaměří na to, zda objekty patřící do skupiny dostávají informace o změně provozního režimu a řídicí povely.
- V třetí řadě se již ověří správná funkčnost programu, tedy splnění provozních pokynů a navržené struktury technologického řízení.

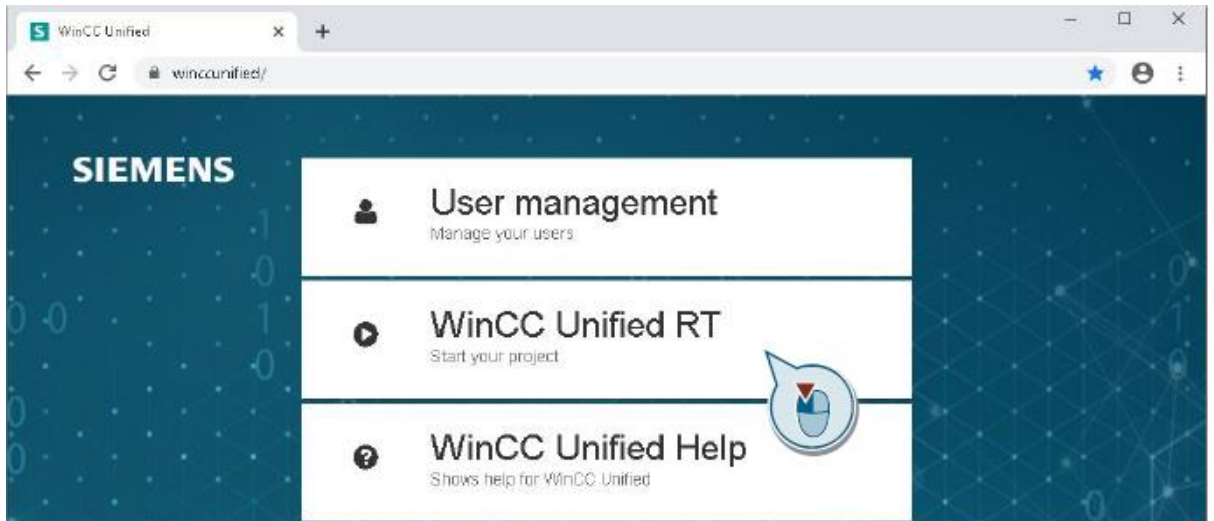
#### 3.1. Zprovoznění simulátoru obrazovek

Využití simulátoru na panelech Unified je odlišné oproti předchozím řadám panelů SIMATIC. Změna spočívá zejména v tom, že na panelech řady Comfort simulace probíhala v popředí TIA Portálu po spuštění simulátoru. Naproti tomu, u panelů Unified probíhá simulace přes lokální webovou doménu "https://localhost".[16][18] Na obrázku č. 39 je k vidění úvodní obrazovka tohoto webového rozhraní.

---

<sup>7</sup> Nástroj přímo v prostředí TIA Portálu umožňující sledovat a modifikovat stavy signálů typu M a proměnných obsažených v datových blocích. Při absenci karet v hardwarové sestavě definujících přístup adresového rámce ze vstupních periférií umožňuje nastavovat hodnoty i vstupním proměnným

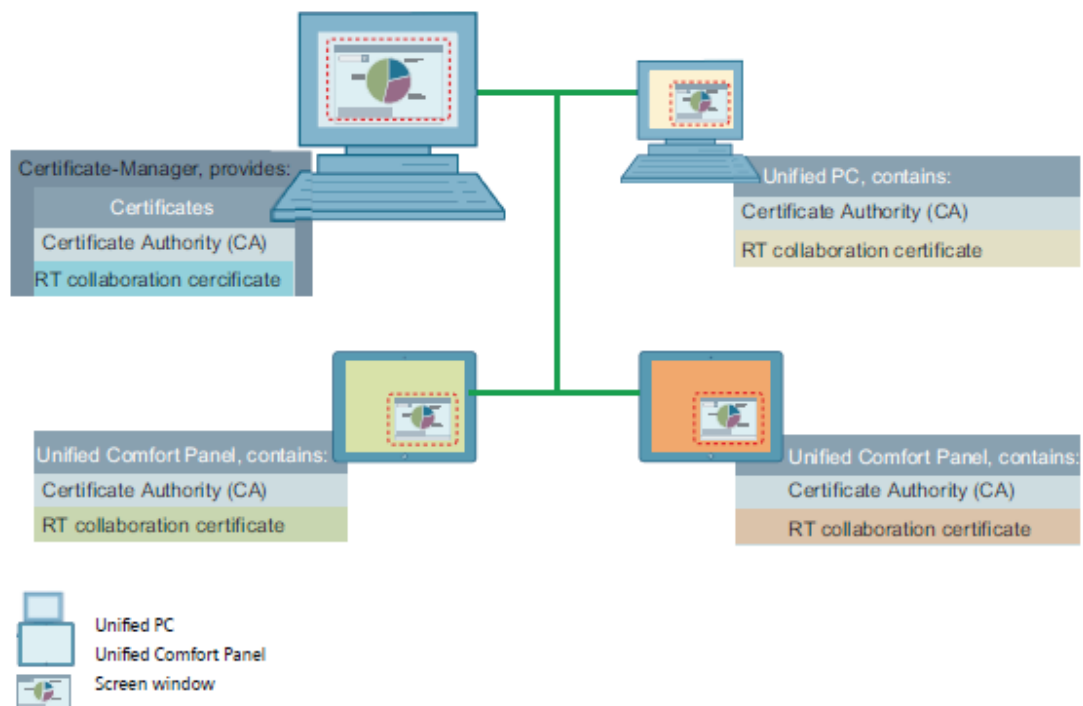
<sup>8</sup> Lépe řečeno, šlo by vstupy modifikovat na jinou úroveň, avšak hned v následujícím programovém cyklu by došlo k jejich přepsání na výchozí hodnoty a nic by se nestalo



Obr. 39: Webové rozhraní po spuštění simulátoru [18]

Pro spuštění simulace, a tedy i režimu RunTime (RT) na panelu Unified, je třeba mít vytvořeného uživatele v simulovaném projektu. Jakmile se operátor přihlásí prostřednictvím uživatelského účtu, může posléze využívat všechny funkce, na které má jeho účet oprávnění [16].

Pro úspěšné zprovoznění simulace bylo třeba vytvořit přes WinCC Unified manager takzvaný self-singed certifikát spojený s IP adresou počítače. Certifikát totiž slouží k ověření, zda připojované zařízení patří mezi důvěryhodné komunikační partnery. Obrázek ilustrující propojení zařízení s webovým serverem je na obrázku 40 níže.



Obr. 40: Propojení zařízení [16]

### 3.2. Kontrola provázaných signálů pro vizualizaci

Skutečnost, že jsou správně propojeny vizualizační objekty se všemi požadovanými signály z PLC lze nejnázorněji ověřit právě pomocí dříve zmíněné simulace. Provádět totiž kontrolu ze samotného programu by bylo jak zdlouhavé, tak i ne příliš spolehlivé. Mohlo totiž dojít k překlepům, systematickým chybám z kopírování, či pouze chybám z nepozornosti. Takovéto chyby by pak mohly být takřka kdekoliv, ať v samotném provázání ve vizualizaci, tak třeba v textových listech, špatně napojeném tagu a tak dále.

Kontrola ve své podstatě probíhala stylem, kdy bylo nejdříve letmo ověřeno správné propojení vyskakovacích oken s textovými listy, které určují popis a název objektu. Důvod tohoto přístupu byl ten, že styl propojování textových listů je shodný se strukturou provázaných tagů. Došlo-li tedy někde k rozdílu v popisu či názvu objektu vůči očekávání, znamenalo to, že nastala jedna ze dvou potenciálních chyb v propojení, tedy:

- Bylo nastavené špatné číslo objektu u vyskakovacího okna.
- Požadovaný objekt byl špatně popsán v textovém listu.

Další chybou, která se vzhledem k volené struktuře vyskytovala, bylo provázání tagů na vstupu instance faceplatu, které odkazovaly na datový blok jiného objektu. Jejím projevem byla například špatná reakce vizualizace na stav chodu motoru, jelikož objekt četl stavy z adresního prostoru jiného objektu.

V programu se nejčastěji vyskytovala kombinace posledních dvou chyb způsobená právě chybou postupu kopírování a vkládání, kde se u některých objektů zapomnělo upravit jejich provázání.

### 3.3. Příprava ověření programové funkce

Aby šlo ověřit propojení mezi technologickými objekty a objektem skupiny, nestačí již simulovat pouze HMI panel, ale je potřeba simulovat i program PLC, jelikož právě v něm je požadovaná vazba implementována. Je tedy třeba nahrát program buď do simulátoru PLC SIM, nebo PLC SIM Advanced, kde PLC SIM Advanced nabízí nesporně řadu výhod, jako například komunikace s externími procesy, avšak navzdory výhodám byl pro účely ověřování vazeb a následného testování programové funkce upřednostněn simulátor PLC SIM. Důvody pro tuto volbu budou upřesněny hned v následujícím odstavci.

Po spuštění obou simulačních procesů však hned vyvstává otázka, jak simulovat vstupy z periférií. Po přidání hardwarové sestavy do projektu již nestačí pouze vstupní signály simulovat pomocí nástroje watch table, nýbrž je potřebná možnost simulace periferních vstupů, která je k dispozici například v rozšířeném náhledu simulátoru PLC SIM, ve kterém jsou po importu tagů z projektu nastavené požadované hodnoty vybraných vstupních signálů. Pro úplnost je vhodné zmínit, že simulátor PLC SIM Advanced rovněž umožňuje simulaci periférií, avšak je na to zapotřebí stáhnout dodatečný nástroj PLC SIM Advanced Simulation Table. Pro účely tohoto projektu bylo se zmíněným nástrojem úspěšně navázáno spojení s programem z PLC, avšak nikdy se ho nepodařilo řádně spustit v režimu umožňujícím právě simulaci periférií.

Dalším nezbytným krokem k přípravě ověření je úprava inicializačních funkcí technologických objektů pro účely simulace. Například objekty motorů totiž obsahují třeba komunikaci s frekvenčními měniči, nebo vstupy lokálního řízení, které nejsou pro účely simulace zajímavé, či jednoduše realizovatelné. Pro zjednodušení práce potřebné k ověření simulací budou tedy deaktivovány všechny signály, které nejsou bezprostředně třeba k uvedení bloku motoru do stavu



## VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

chodu. Předpokladem pro takovéto zjednodušení je skutečnost, že zanedbané signály mohou do chodu programu zavést chybu akorát při záměně napojeného signálu, nebo se netýkají řídicího programu a slouží k vnitřnímu vyhodnocování ve funkčním bloku. V obou případech platí, že funkce firemně standardizovaných bloků základních objektů je odzkoušena léty praxe a vývoje. Díky této skutečnosti a zmíněnému zjednodušení se lze dále zaměřit pouze na ověření správné implementace programových vazeb a technologického procesu.

V inicializačních funkcích objektů pro simulační účely jsou po úpravách ponechány pouze nezbytně nutné funkce pro správný chod a u objektů analogového a digitálního měření zapnut simulační režim. Jde o režim, který nabízí firemně standardizované funkční bloky a umožňuje ignorovat vstupní hodnoty měřených signálů, a místo toho nastavovat simulovanou měřenou hodnotu přímo z vizualizace. Zmíněná funkce bude proto široce využívána pro ulehčení práce ve zbytku kapitoly zabývající se simulací výsledků. Na vizualizačních obrazovkách je simulační režim objektů měření znázorněn žlutým podbarvením okolní oblasti.

### 3.4. Testování implementované technologické funkce

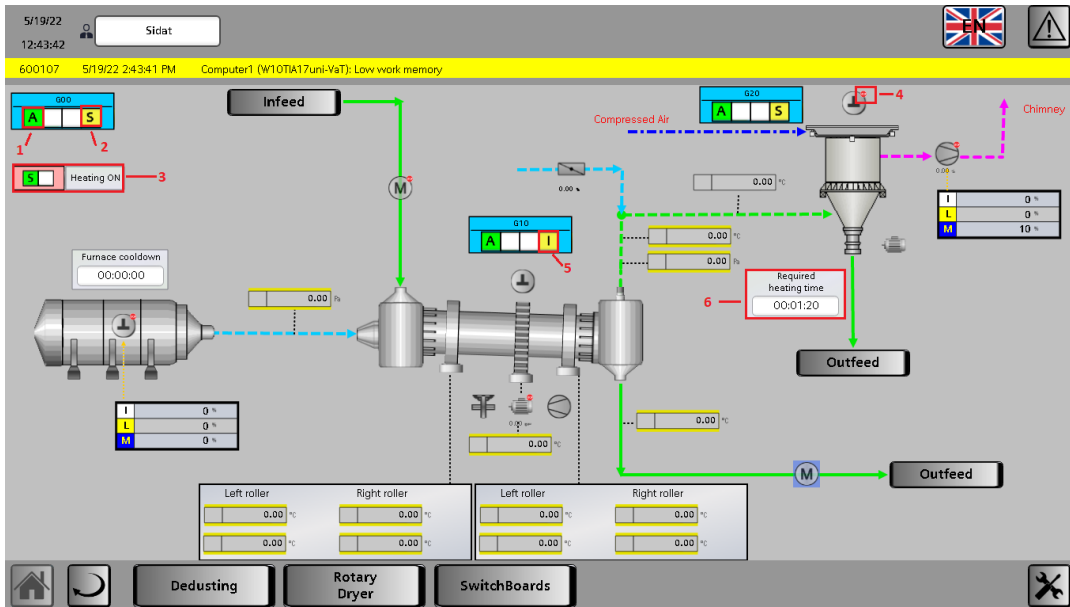
S dokončeným řídicím PLC programem, vytvořenou vizualizací a ověřením provázaných signálů přichází na řadu testování technologie jako celku. Tedy ověření programových vazeb, zda je správně nastavené technologické zapínání, vypínání a provoz linky dle pokynů výrobců a technologa.

Funkci simulovaného programu však bohužel nelze jednoduše demonstrovat v rámci písemné zprávy. Hlavním důvodem je to, že by bylo nutné velké množství obrázků, na kterých by byly vidět jednotlivé kroky zapínání a vypínání technologických objektů. S ohledem na tuto skutečnost budou znázorněny a popsány pouze vybrané stavy procesu a v dodatcích dokumentu bude k nalezení odkaz na video demonstrující základní funkce.

Strategie testů byla zvolena tak, že budou nejdříve otestovány jednotlivé sekce linky samostatně z hlediska technologického zapínání, vypínání a blokačních vazeb. Následně se věnuje pozornost situacím, ve kterých došlo například k dosažení kritické hodnoty měření, či selhání některé z technologických částí linky, a jestli je reakce řízení zbytku linky adekvátní. Hlavní soupis požadované funkce jednotlivých částí je v kapitolách [Popis řízení filtru](#) a [Popis řízení rotační sušárny](#).

Prvním krokem simulování je zapnutí filtrační sekce, neb lze jako jediná zapnout plně nezávisle na zbytku linky. Dle technologického popisu by zapínání sekce mělo jít v pořadí ohřev, pohon klapky, řídicí jednotka regeneračního cyklu a naposled ventilátor. Demonstrací spouštění sekce se budou zabývat následující tři obrázky. Prvním je obrázek číslo 41 níže, ten obsahuje zvýrazněné prvky zájmu, které budou pod ním popsány. Mimo popis se jedná o první krok zapnutí filtrační linky, kdy je nejprve nutné zajistit, že alespoň dvě hodiny před zpuštěním filtru byl zapnutý ohřev násypky. Pro účely simulace jsou pro urychlení všechny doby čekání zkráceny.

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

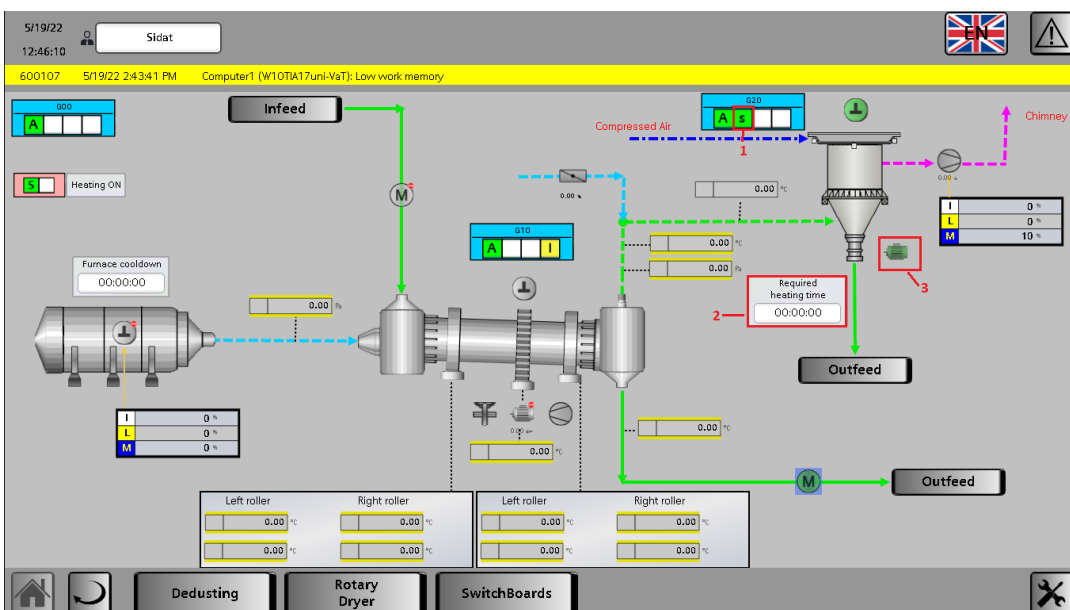


Obr. 41: Přehledová obrazovka linky s přidávanými číselnými odkazy na popis

Význam číselných odkazů z obrázku 41 výše je následovný:

1. Skupina je v automatickém módu
2. Skupina má aktivovanou blokáci startu
3. Je zapnutý ohřev násypky filtru
4. Řídící jednotka regeneračního cyklu filtru je ve stavu s aktivovanou blokáci
5. Skupina má aktivovanou provozní blokáci
6. Zbývající nutný čas ohřevu násypky před možným spuštěním filtru

Další obrázek, tedy obrázek číslo 42 již demonstruje situaci po ukončení doby čekání na ohřev a zadání pokynu ke spuštění skupiny filtru. Lze na něm zároveň vidět již spuštěný pohon dvojité výkyvné klapky filtru a řídicí jednotku regeneračního cyklu v chodu (indikováno zelenou barvou).



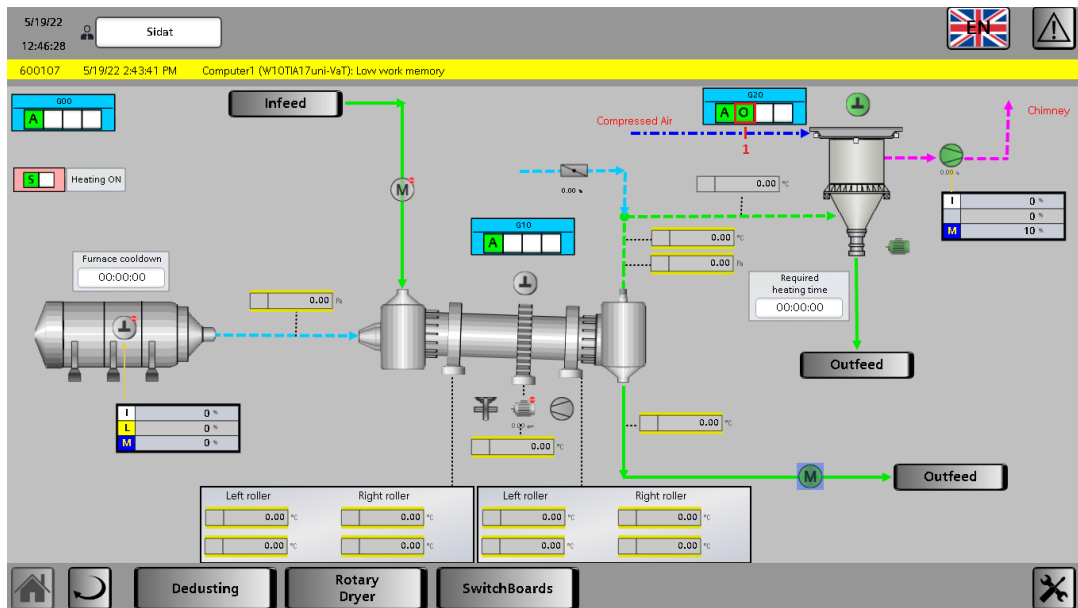
VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

Obr. 42: Přehledová obrazovka linky během zapínání filtru s přidánými číselnými odkazy na popis

Význam číselných odkazů z obrázku 42 výše je následovný:

1. Skupina je ve stavu startování
2. Uplynul potřebný čas ohřevu, a tedy skupina již není déle ve startovací blokaci
3. Pohon výkyvné dvojité klapky filtru je aktivní (symbolizováno zelenou barvou).

Navazující obrázek 43 již demonstruje situaci, kdy bylo ukončeno zapínání skupiny filtru, a tedy i odblokována skupina rotační sušárny.



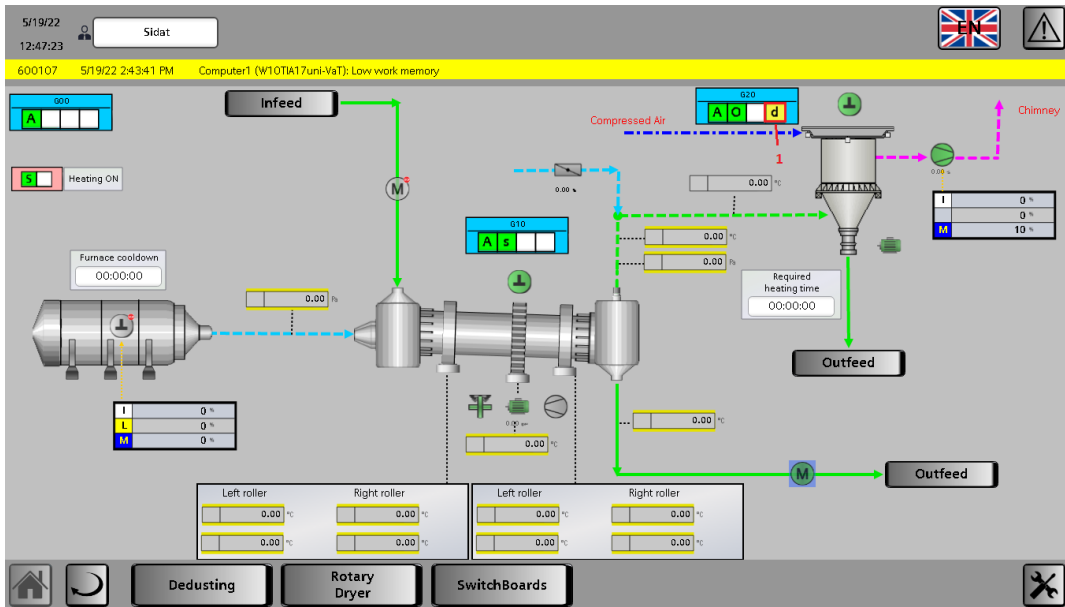
Obr. 43: Přehledová obrazovka linky po zapnutí filtru s přidánými číselnými odkazy na popis

Význam číselných odkazů z obrázku 43 výše je následovný:

1. Skupina obdržela zpětné hlášky řízených objektů, ukončila startovací fázi a je již pouze ve stavu chodu.

Následující tři obrázky znázorňují spouštění technologické sekce rotační sušárny. Zde je dle technologického popisu očekáváno zapínání technologických v pořadí: řídicí jednotka mazání ozubeného věnce, odjištění brzdy pohonu bubnu, zapnutí pohonu bubnu, spuštění chladicího ventilátoru pohonu, spuštění řídicí jednotky spalovací komory, a nakonec nastavit signál připravenosti k přísunu materiálu, pokud teplota výstupního vzduchu sušárny překročí 80 °C. Obrázek číslo 44 níže demonstruje první část zapínání sekce rotační sušárny.

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

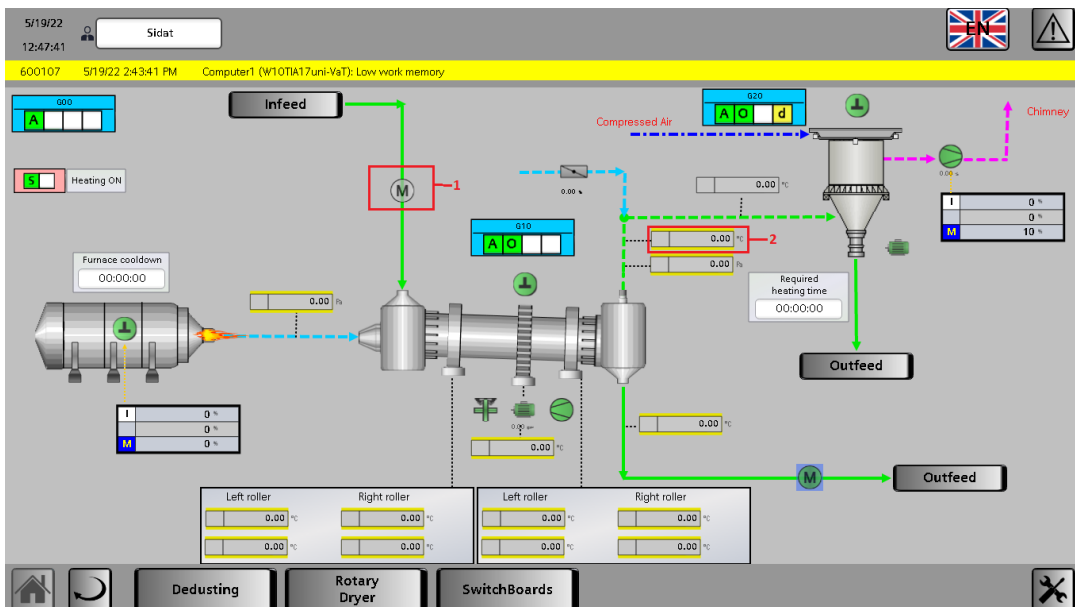


Obr. 44: Přehledová obrazovka linky během spuštění sekce rotační sušárny

Význam číselných odkazů z obrázku 44 výše je následovný:

1. Skupina je blokována pro vypnutí. Dokud není odblokována, nelze vypnout povelom od operátora.

Obrázek číslo 45 níže demonstruje situaci, kdy je sice skupina rotační sušárny již spuštěná, avšak stále nenastavila signál připravenosti k přísunu materiálu kvůli nízké teplotě výstupního vzduchu z bubny sušárny. Signál o připravenosti je nastaven až na obrázku 46, kde lze právě i vidět dostatečně vysokou nastavenou teplotu výstupního vzduchu.

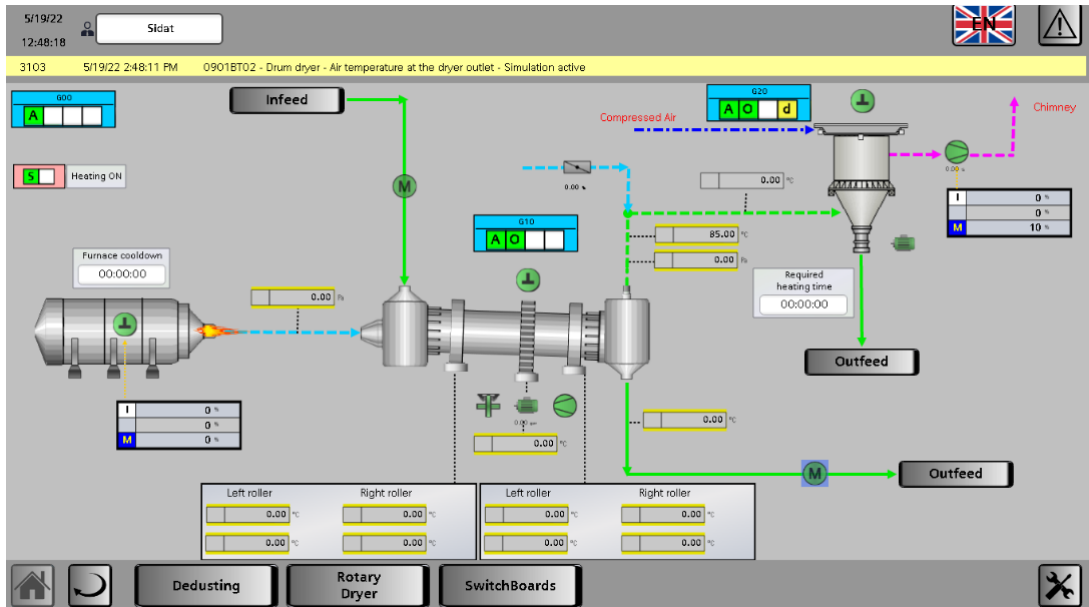


Obr. 45: Přehledová obrazovka linky při zapínání části rotační sušárny číselnými odkazy na popis

Význam číselných odkazů z obrázku 45 výše je následovný:

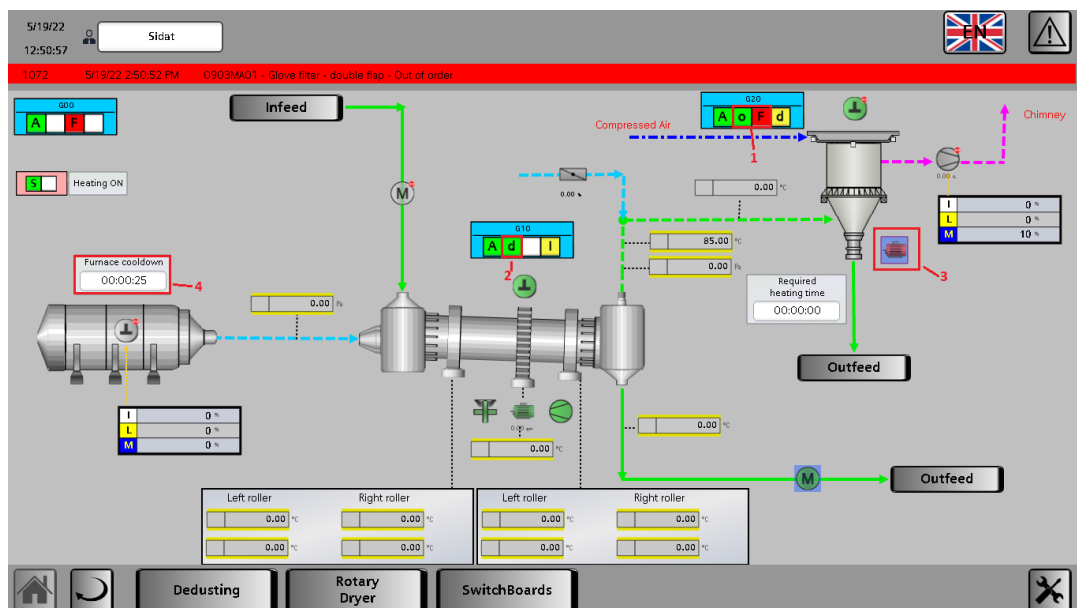
VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

1. Provizorní ikona pro značení signálu sušárny o připravenosti k přísunu materiálu. Je neaktivní do doby, než teplota výstupního vzduchu sušárny (bod 2) překročí 80 °C.
2. Teplota výstupního vzduchu sušárny. Jakmile překročí 80 °C, odblokuje objekt z bodu 1.



Obr. 46: Přehledová obrazovka linky po splnění podmínek z obrázku 45

Následující dva obrázky, číselně 47 a 48, demonstrují reakci linky na poruchu pohonu výkyvné dvojité klapky filtru. První obrázek 47. znázorňuje a popisuje situaci, během které teprve dochází k vypnutí, kdežto obrázek 48 zobrazuje linku již po ukončení vypínací sekvence. Dle technologického popisu má na chybu reagovat filtrační jednotka, která zajistí technologické odstavení skupiny, důsledkem čehož dojde i k vypnutí ventilátoru. Ventilátor filtru je však základní podmínkou chodu rotační sušárny, díky čemuž dojde i k jejímu vypnutí dle dříve popsaných specifikací.

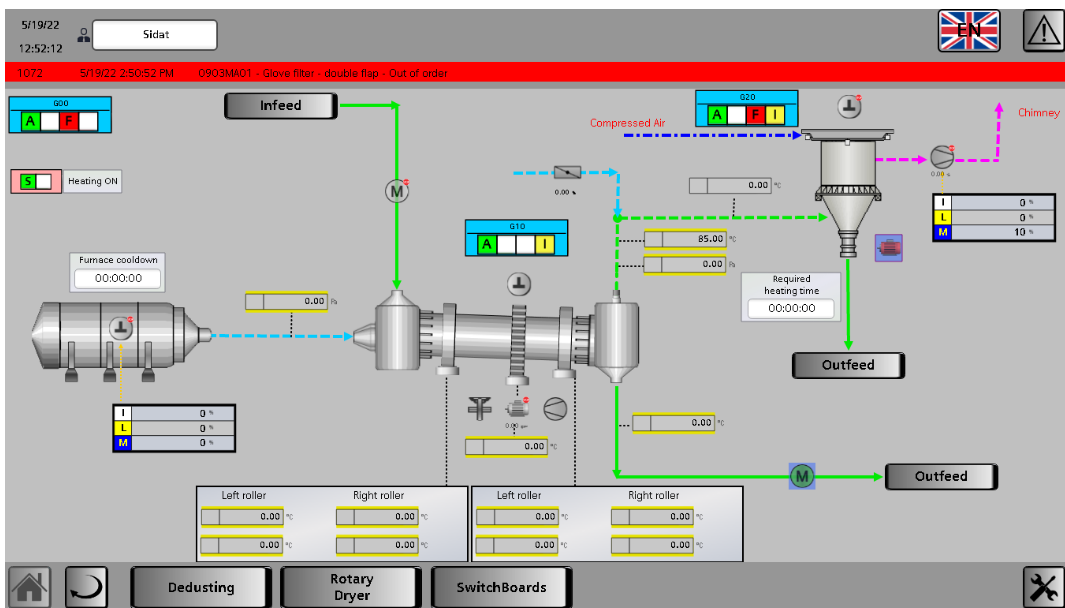


Obr. 47: Reakce linky na chybu pohonu výkyvné dvojité klapky

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

Význam číselných odkazů z obrázku 47 výše je následovný:

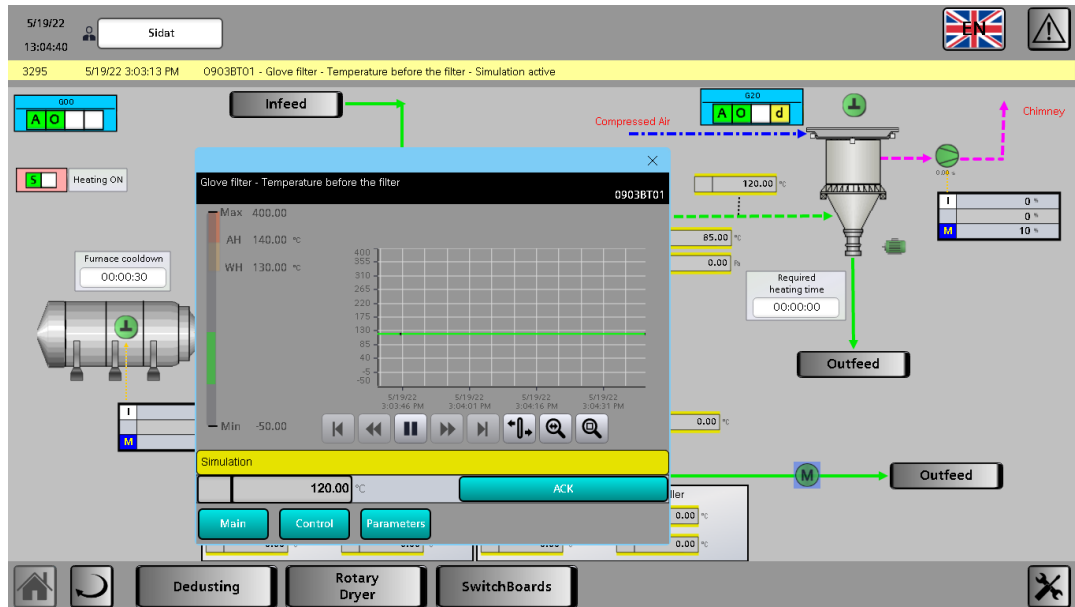
1. Skupina filtru ztratila zpětné hlášení a je v chybě. Dále na základě pokynů individuálních objektů dojde k technologickému vypnutí filtrační jednotky dle specifikací
2. Skupina rotační sušárny přešla kvůli operační blokaci do stavu vypínání. Vypínání je technologické dle specifikací, avšak jelikož došlo k vypnutí ventilátoru, je spalovací komora odstavena okamžitě bez dodatečného zpoždění.
3. Pohon je v chybě (značeno červeným blikáním) a v manuálním režimu (značeno modrým okolním polem)
4. Čas chlazení spalovací komory. Nejedná se o čas do vypnutí, ale o čas, za který je nejdříve možné znovu spustit rotační sušárnu. Jedná se o požadavek technologa.



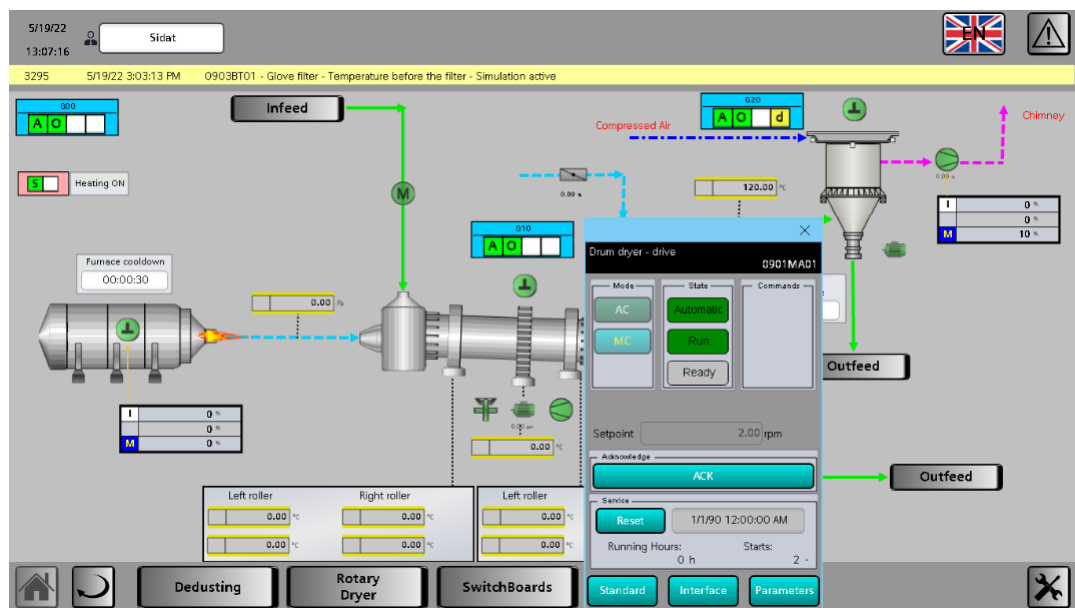
Obr. 48: Vypnutá linka po reakci na chybu pohonu klapky filtru

Na následujících obrázcích 49, 50 a 51 je ukázána struktura vyskakovacího okna objektu analogu, objektu motoru a další záložka stejného objektu motoru. Obrázky mají, jak demonstrovat funkčnost, tak i dát čtenáři představu o obsahu vyskakovacích oken, neb krom slovního popisu o tvorbě a obsahu okna nebyl náhled na vizuální podobu za chodu v práci nikde uveden, jelikož se jedná o funkci převzatou z vizualizačního standardu firmy.

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

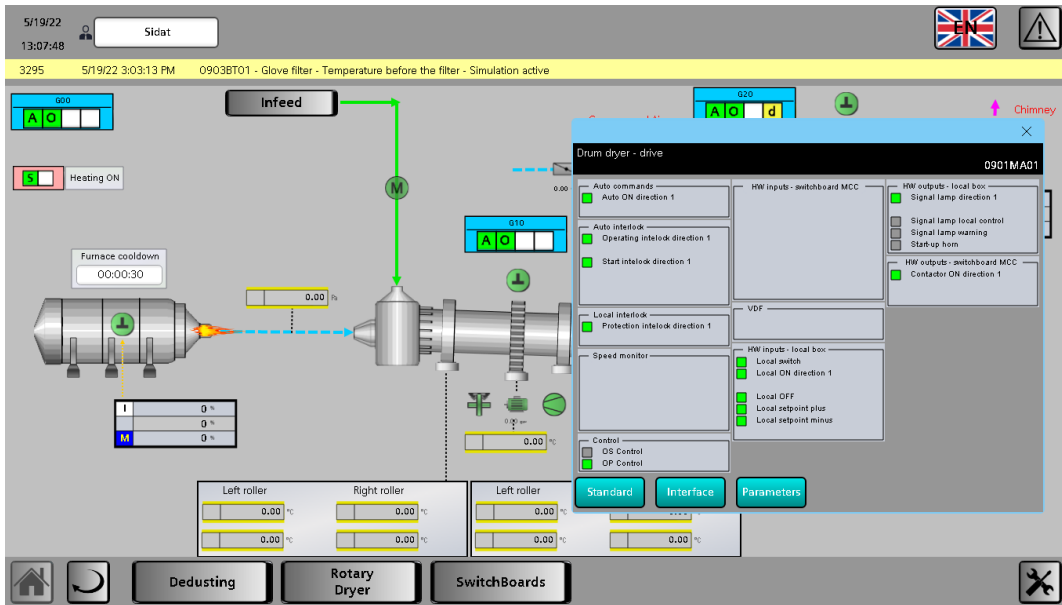


Obr. 49: Vyskakovací okno analogového objektu s průběhem grafu hodnot a znázorněním mezních rozsahů varování a alarmů



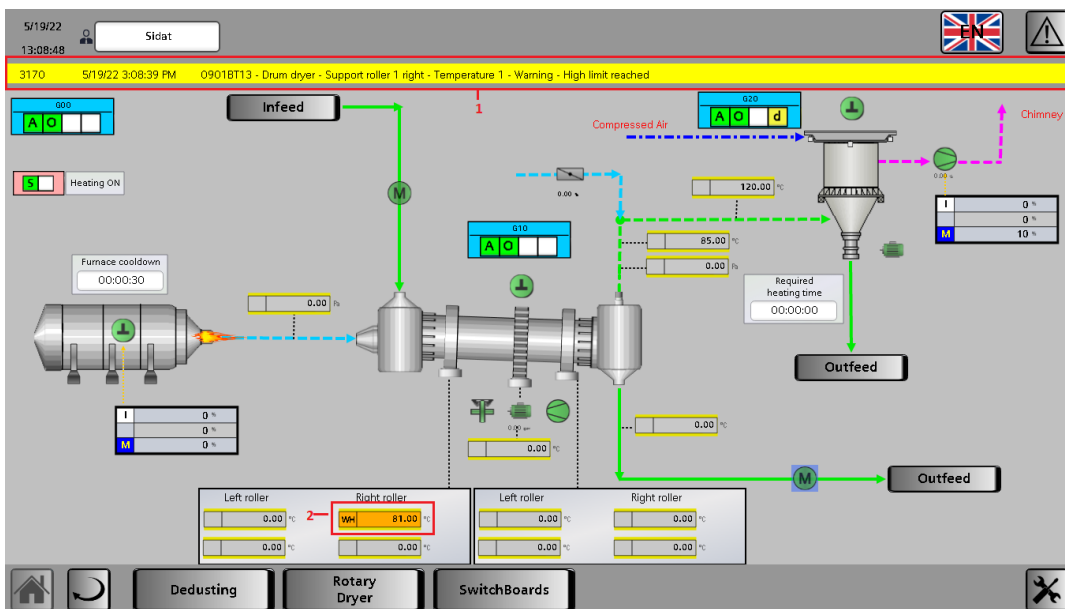
Obr. 50: Základní obrazovka vyskakovacího okna pohonu bubnu sušárny za simulovaného provozu

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE



Obr. 51: Záložka „interface“ vyskakovacího okna pohonu za simulovaného provozu linky

Následující čtyři obrázky demonstrují reakci na překročení varovné hladiny a alarmové hladiny. Vizuální stránku překročení varovné hladiny měření spolu s varovací hláškou znázorňuje obrázek 52. Následuje obrázek 53, kde je již překročena alarmová (kritická) hladina měření a dochází k vypnutí sekce linky dle popisu technologie. Na obrázku 54 je již skupina rotační sušárny v reakci na překročení povoleného rozsahu měřeného signálu vypnutá. Obrázek číslo 55 tuto demonstraci uzavírá zobrazením detailu alarmů, spolu se záznamem dříve vzniklých alarmových hlášek.



Obr. 52: Náhled na reakci na aktivaci varovné hladiny analogového měření teploty podpěrného válečku sušárny

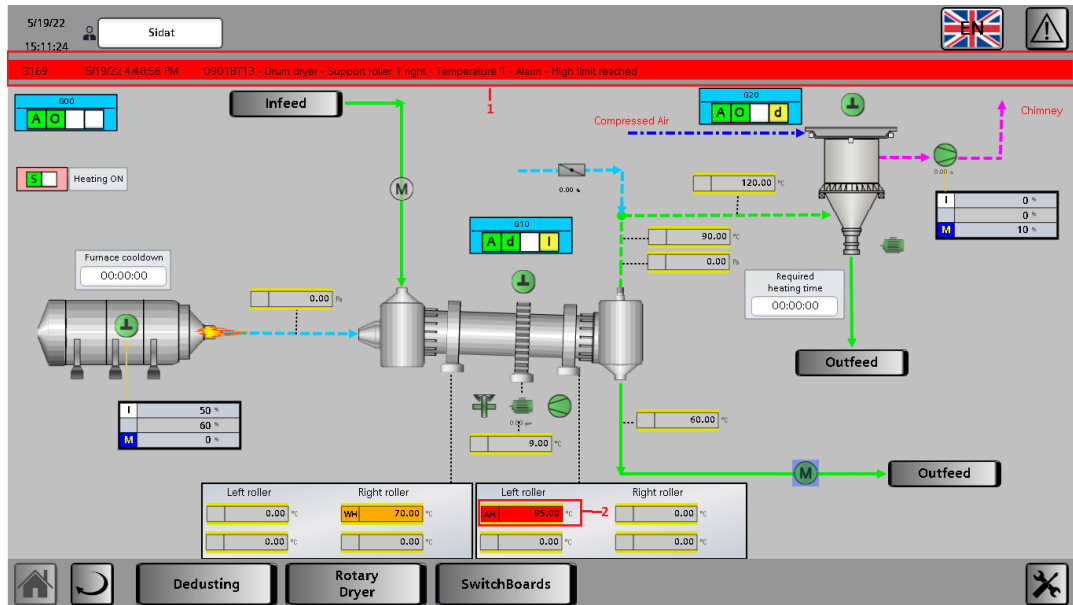
Význam číselných odkazů z obrázku 52 výše je následovný:

1. Zobrazení varovací hlášky upozorňující na problém



VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

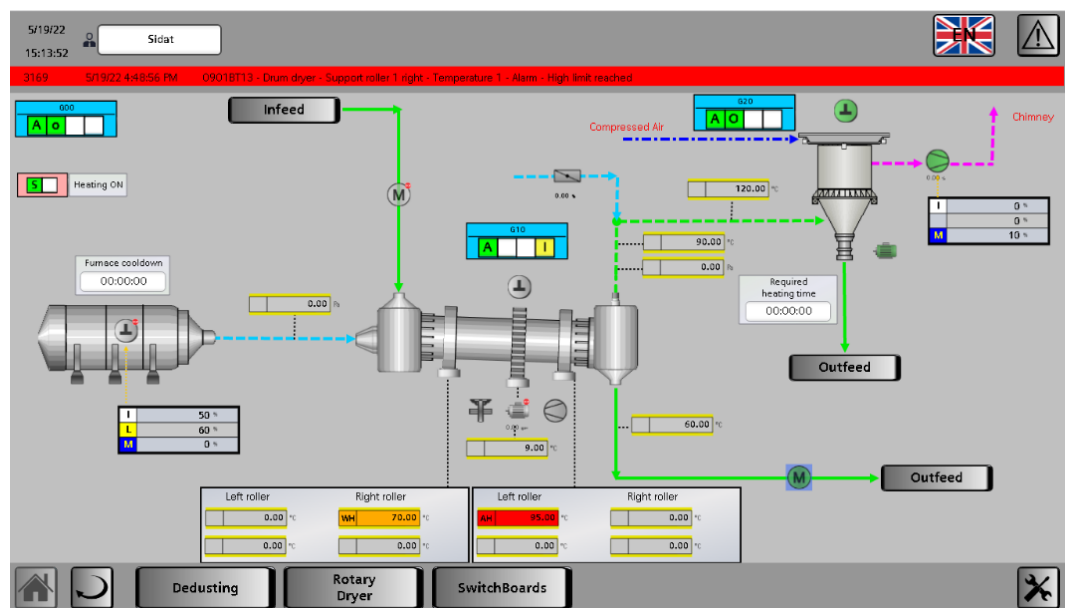
2. Vizuální indikace překročení varovné meze analogového měření teploty podpěrného válečku sušárny



Obr. 53: Náhled na reakci na aktivaci alarmové hladiny analogového měření teploty podpěrného válečku bubnu sušárny

Význam číselných odkazů z obrázku 53 výše je následovný:

1. Zobrazení alarmové hlášky upozorňující na problém
2. Vizuální indikace překročení alarmové meze analogového měření teploty podpěrného válečku bubnu sušárny. Na všechny alarmové meze je naprogramovaná i adekvátní reakce linky. V tomto konkrétním případě dojde k technologickému vypnutí sekce rotační sušárny.



Obr. 54: Náhled po vypnutí sekce rotační sušárny v reakci na překročení alarmové meze analogového měření teploty podpěrného válečku bubnu sušárny

VIZUALIZACE TECHNOLOGICKÉ PROCESU SUŠÁRNY VÁPENCE

ID	Alarm class	Alarm text	Raise time	Status text
3169	S_Alarm	0901BT13 - Drum dryer - Support roller 1 right - Temperature 1 - Alarm - High limit reached	5/19/22 4:48:56 PM	Incoming
25	S_Alarm	MCC01-93QC01 - Breaker - Alarm	5/18/22 4:52:30 PM	Incoming
26	S_Alarm	MCC01-93FA01_02 - Breaker - Alarm	5/18/22 4:52:30 PM	Incoming
27	S_Alarm	MCC01-95FA01 - Breaker - Alarm	5/18/22 4:52:30 PM	Incoming
28	S_Alarm	MCC01-10FA03_04 - Breaker - Alarm	5/18/22 4:52:30 PM	Incoming
29	S_Alarm	0917RT01QM01 - Electric hoist - Securing - Alarm	5/18/22 4:52:30 PM	Incoming
30	S_Simulation	0901BT02 - Drum dryer - Air temperature at the dryer outlet - Simulation active	5/19/22 2:48:11 PM	Incoming
31	S_Simulation	0903BT01 - Glove filter - Temperature before the filter - Simulation active	5/19/22 3:03:13 PM	Incoming
32	S_Warning	0901BT13 - Drum dryer - Support roller 1 right - Temperature 1 - Warning - High limit reached	5/19/22 3:08:39 PM	Incoming
33	S_Alarm	0901BT21 - Drum dryer - Support roller 2 left - Temperature 1 - Alarm - High limit reached	5/19/22 3:09:20 PM	Incoming
34	S_Warning	0901BT21 - Drum dryer - Support roller 2 left - Temperature 1 - Warning - High limit reached	5/19/22 3:09:20 PM	Incoming
35	SystemAlarm	Computer1 (W10TA17uni-VaT): Very low hard disk space	5/19/22 3:37:57 PM	Incoming/Acknowledg
36	SystemWarn	Computer1 (W10TA17uni-VaT): Low hard disk space	5/19/22 3:37:57 PM	Incoming/Acknowledg
37	S_Simulation	0904BT02 - Filter fan - Bearing temperature 2 - Simulation active	5/19/22 4:45:25 PM	Incoming
38	S_Warning	0904BT02 - Filter fan - Bearing temperature 2 - Warning - High limit reached	5/19/22 4:45:25 PM	Incoming
39	S_Alarm	0904BT02 - Filter fan - Bearing temperature 2 - Alarm - High limit reached	5/19/22 4:45:36 PM	Incoming
40	S_Alarm	0901BT13 - Drum dryer - Support roller 1 right - Temperature 1 - Alarm - High limit reached	5/19/22 4:48:56 PM	Incoming

Obr. 55: Detail alarmových hlášek po vzniku chyby

Po důkladném testování propojení jednotlivých částí a opravení případných chyb byla sada testů zakončena ověřením celkové programové vazby, při kterých bylo testováno hromadné zapínání, vypínání a zda je celková reakce na možné poruchy adekvátní. Po skončení všech testů se již v projektu nevyskytují chyby typu nedodržení technologického procesu, alespoň z hlediska možného ověření pomocí simulace.

## 4. Závěr

V rámci práce byl jak dle zadání, tak i dle potřeby nastudován a popsán výrobní proces sušení vápence spolu s podklady nutnými pro práci v prostředí TIA Portal a WinCC Unified.

Pro zakázku byl na základě podkladů obdržených od zákazníka, výrobce a technologa zkompletován návrh řízení, nutný pro správnou funkci výrobní linky, který zároveň splňuje specifikace výrobců filtrační jednotky a rotační sušárny. Navržené řízení bylo posléze implementováno do PLC programu, avšak jelikož nebyl krok přímo součástí zadání, byl v práci rozebrán spíše stručně z hlediska programového strukturování.

Na tvorbu programu navázal popis implementace vizualizačních obrazovek na operátorské panely Unified, spolu s popisem prozkoumávaných metod implementace nového firemního vizualizačního standardu. Následně je vysvětlena struktura převzatého vyvíjeného vizualizačního standardu a s jeho využitím navržena podoba vizualizačních obrazovek, které jsou zároveň navázány na data z programu PLC.

Po dokončení jak programové, tak vizualizační části byl výsledek práce odzkoušen a otestován pomocí simulátorů z hlediska reakce na řídicí signály, chyby a celkového propojení. Je však důležité mít stále na paměti, že simulace sice umožní odladit spoustu chyb a nedodělků, avšak nelze s jejím využitím odladit vše. Nicméně stále platí, že návrh, realizace i ověřování byly provedeny s co nejsvědomitějším přístupem.

Na práci mělo být posléze původně navázáno zprovozněním a otestováním na fyzické lince pro zákazníka v Rusku, avšak díky změně politické situace mezi zadáním a odevzdáním práce byl tento krok odložen na dobu neurčitou. V každém případě bude stále nutné na projektu pokračovat, stále vyjasnit dotazy ohledně některých technologických vazeb, dokončit detaily na vizualizačních obrazovkách a upravit případné vazby dle dodatečných požadavků zákazníka. Při případném fyzickém zprovozněním bude stále nutné naladit regulátory a otestovat programové vazby při reálném zapojení.

## Dodatky

- a. Význam signálů uvedených na obrázcích v kapitole 2.2.4 je uveden v dodatkové tabulce číslo 1.

Tab. 1: Vysvětlivky signálů z kapitoly 2.2.4

Objekt	Význam
09G20	Řídící skupina filtru
09G21	Podskupina filtru
0903MA01	Pohon výkyvné dvojité klapky filtru
0903RT01	Řídící jednotka regeneračního cyklu filtru
0903EH01	Ohřev násypky filtru
0904MA01	Ventilátor filtru
0903RT01ST01	Termostat ohřevu násypky filtru
0903RT01SL01	Nižší maximální senzor hladiny násypky filtru
0903RT01SL02	Vyšší maximální senzor hladiny násypky filtru
0903RT01BP01	Pokles tlaku filtru

- b. Odkaz na video demonstrující funkci vizualizace na videu bez editace, popisů či komentáře. Pouze 2x zrychlené vůči reálnému času:

<https://youtu.be/aLImHSkOTLo>

## Seznam literatury

- [1] CAPITAINE, Shane Le. FEECO INTERNATIONAL. *A Detailed Look At Limestone Drying* [online]. [cit. 2022-03-08]. Dostupné z: <https://feeco.com/dry-limestone>
- [2] KOZICKI, Chris. *Benefits Of Pelletizing Limestone* [online]. Feeco international [cit. 2022-03-08]. Dostupné z: <https://feeco.com/benefits-pelletizing-limestone>
- [3] CAPITAINE, Shane Le. FEECO INTERNATIONAL. *8 Benefits of Using a Rotary Dryer for Drying Bulk Solids* [online]. Process Heating, 2016 [cit. 2022-03-08]. Dostupné z: <https://www.process-heating.com/articles/91538-benefits-of-using-a-rotary-dryer-for-drying-bulk-solids>
- [4] KOZICKI, Chris. *Pelletized Lime Vs. Ag Lime (And Other Imposters)* [online]. Feeco international [cit. 2022-03-08]. Dostupné z: <https://feeco.com/pelletized-lime-vs-ag-lime-imposters>
- [5] Castaño, Fernando & Jimenez, D. & Rubio, Francisco. (2004). *A combined  $H^\infty$ /QFT control of a rotary dryer*. 6. 5909 - 5914 Vol.6. 10.1109/CDC.2003.1271950.
- [6] VERTICAL BAG FILTERS [online]. INSTAL-FILTER [cit. 2022-05-20]. Dostupné z: <https://instalfilter.com/Vertical-bag-filters.html>
- [7] SIMATIC STEP 7 Basic/Professional V17 and SIMATIC WinCC V17 [online]. Siemens, 2021 [cit. 2022-01-08]. Dostupné z: <https://support.industry.siemens.com/cs/document/109798671/simatic-step-7-basic-professional-v17-and-simatic-wincc-v17?dti=0&lc=en-WW>
- [8] *Programming Guideline for S7-1200/S7-1500* [online]. Siemens, 2017 [cit. 2022-01-08]. Dostupné z: <https://support.industry.siemens.com/cs/document/90885040/programming-guideline-for-s7-1200-s7-1500?dti=0&lc=en-WW>
- [9] *Types of Interrupt OB in Siemens Tia Portal* [online]. Inst Tools [cit. 2022-05-10]. Dostupné z: <https://instrumentationtools.com/types-of-interrupt-ob-in-siemens-tia-portal/>
- [10] *Little/Big endian* [online]. TechTarget, 2021 [cit. 2022-01-08]. Dostupné z: <https://www.techtarget.com/searchnetworking/definition/big-endian-and-little-endian>
- [11] VOJÁČEK, Antonín. *Problematika Modbus přenosu 32bitových hodnot* [online]. HW group, 2018 [cit. 2022-01-08]. Dostupné z: <https://automatizace.hw.cz/problematika-modbus-tcp-prenosu-32bitovych-hodnot.html>

- [12] *What is HMI?*[online]. inductiveautomation, 2018 [cit. 2022-01-08]. Dostupné z: <https://www.inductiveautomation.com/resources/article/what-is-hmi>
- [13] "IEEE Standard for SCADA and Automation Systems," in *IEEE Std C37.1-2007 (Revision of IEEE Std C37.1-1994)*, vol., no., pp.1-143, 8 May 2008, doi: 10.1109/IEEESTD.2008.4518930.
- [14] *Hmi\_unified\_comfort\_panels\_operating\_instructions\_enUS\_en-US*[online]. Siemens, 2020 [cit. 2022-01-08]. Dostupné z: <https://support.industry.siemens.com/cs/document/109773257/simatic-hmi-hmi-devices-unified-comfort-panels?dti=0&lc=en-CZ>
- [15] *Highlights of the new SIMATIC HMI Unified Comfort Panels* [online]. Siemens [cit. 2022-05-10]. Dostupné z: <https://new.siemens.com/global/en/products/automation/simatic-hmi/wincc-unified/hardware.html>
- [16] *SIMATIC HMI WinCC Unified WinCC Engineering V17 – WinCC Unified*[online]. Siemens, 2021 [cit. 2021-09-29]. Dostupné z: <https://support.industry.siemens.com/cs/document/109794204/simatic-hmi-wincc-unified-wincc-engineering-v17-%E2%80%93-wincc-unified?dti=0&lc=en-US>
- [17] *SIMATIC HMI WinCC (TIA Portal) WinCC Engineering V17 – Communication* [online]. Siemens, 2021 [cit. 2021-09-29]. 05/2021. Dostupné z: [https://support.industry.siemens.com/cs/document/109794203/simatic-hmi-wincc-\(tia-portal\)-wincc-engineering-v17-%E2%80%93-communication?dti=0&lc=en-US](https://support.industry.siemens.com/cs/document/109794203/simatic-hmi-wincc-(tia-portal)-wincc-engineering-v17-%E2%80%93-communication?dti=0&lc=en-US)
- [18] *Guide for switching from Comfort Panels to Unified Comfort Panels* [online]. Siemens, 2020 [cit. 2022-01-08]. Dostupné z: <https://support.industry.siemens.com/cs/document/109768002/guide-for-switching-from-comfort-panels-to-unified-comfort-panels?dti=0&lc=en-US>
- [19] *CPU-CPU Communication with SIMATIC Controllers* [online]. Siemens, 2013 [cit. 2022-01-08]. Dostupné z: <https://support.industry.siemens.com/cs/document/78028908/cpu-cpu-communication-with-simatic-controllers?dti=0&lc=en-WW>
- [20] *What is Industrial Ethernet?*[online]. Versa Technology, 2021 [cit. 2022-05-10]. Dostupné z: <https://www.versatek.com/what-is-industrial-ethernet/>