



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# Software pro řízení experimentální linky na výrobu nanovláknenných produktů

## Diplomová práce

*Studijní program:* N2612 – Elektrotechnika a informatika  
*Studijní obor:* 3902T005 – Automatické řízení a inženýrská informatika  
*Autor práce:* **Bc. Filip Gudel**  
*Vedoucí práce:* Ing. Martin Diblík, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# Software for control of experimental nanofibre production machine

## Master thesis

*Study programme:* N2612 – Electrical Engineering and Informatics  
*Study branch:* 3902T005 – Automatic Control and Applied Computer Science

*Author:* **Bc. Filip Gaudel**  
*Supervisor:* Ing. Martin Diblík, Ph.D.



## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Filip Gaudel**  
Osobní číslo: **M15000195**  
Studijní program: **N2612 Elektrotechnika a informatika**  
Studijní obor: **Automatické řízení a inženýrská informatika**  
Název tématu: **Software pro řízení experimentální linky na výrobu nanovláknenných produktů**  
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

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

1. Seznamte se s aktuálním mechanickým, elektrickým a programovým řešením experimentální linky na výrobu nanovláknenných produktů.
2. Na základě zjištěných skutečností navrhnete nový software pro řízení linky. Program vhodně strukturujte vzhledem k mechanickému uspořádání linky, navrhnete odpovídající datové struktury pro řízení a pro práci s parametry.
3. Navrhnete nový algoritmus pro řízení stroje v automatickém režimu tak, aby obsluha mohla snadno měnit konfiguraci dle aktuálního mechanického uspořádání a modifikovat parametry produkce.
4. Navrhnete vhodné grafické uživatelské rozhraní s možností ukládání konfigurace a parametrů pro pozdější opakované použití.
5. K vytvořenému softwaru sestavte jednoduchý návod k použití.

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **40–50 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] **JOHN, Karl-Heinz; TIEGELKAMP, Michael. IEC 61131-3: programming industrial automation systems : Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids . 2nd ed.**
- [2] **B+R Automatizace. Automation Training Materials, 2009**
- [3] **MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty. Vyd. 2. V Praze: Vydavatelství ČVUT, 2004. ISBN 80-01-02925-5.**

Vedoucí diplomové práce:

**Ing. Martin Diblík, Ph.D.**

Ústav mechatroniky a technické informatiky

Datum zadání diplomové práce: **10. října 2017**

Termín odevzdání diplomové práce: **14. května 2018**

prof. Ing. Zdeněk Plíva, Ph.D.  
děkan



*Kolář*  
doc. Ing. Milan Kolář, CSc.  
vedoucí ústavu

V Liberci dne 10. října 2017

## Prohlášení

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

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

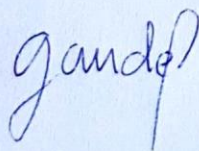
Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

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

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

Datum: 18. 12. 2018

Podpis:



## Poděkování

Touto cestou bych rád poděkoval těm, kteří svou pomocí vědomě či nevědomě přispěli ke vzniku této práce. Zvláště bych rád poděkoval Ing. Martinu Diblíkovi, Ph.D. za trpělivost při vedení této práce.

## Abstrakt

*Software pro řízení experimentální linky na výrobu nanovláknenných produktů.*

Práce se zabývá návrhem softwaru pro řízení experimentální linky na výrobu nanovláknenných produktů a seznamuje s mechanickým a s elektrickým provedením ústrojí. Navržením nového softwaru sloužícího k řízení stroje v automatickém režimu a vytvoření grafického uživatelského rozhraní pro snadnou manipulaci s programem a možností pozdějšího opakovaného použití.

### **Klíčová slova:**

B&R Automation Studio, řízení experimentální linky, výroba nanopříze, výroba nanovláknna.

## Abstract

*Software for control of experimental nanofibre production machine.*

The thesis deals with the design of the software for the management of the experimental line for the production of nanofibrous products and introduces the mechanical and the electrical design. Designing new software to control the machine in automatic mode and create a graphical user interface for easy handling of the program with the possibility of later reuse.

### **Key words:**

B&R Automation Studio, control of experimental line, production of nanopowders, production of nanofibres.

# Obsah

Úvod.....	11
<b>1 B&amp;R Automation .....</b>	<b>12</b>
1.1 Produkty B&R.....	13
1.1.1 Programovatelné logické automaty (System X20, X67) .....	13
1.1.2 Vykonávání programu PLC.....	15
1.2 Komunikační standardy .....	16
1.2.1 Powerlink.....	16
1.2.2 X2X Link.....	16
1.2.3 Profibus, DeviceNet, Can .....	16
<b>2 Automation studio .....</b>	<b>17</b>
2.1 Automation Runtime .....	17
2.2 Vytvoření programu a pracovní prostředí .....	17
2.3 Práce se simulátorem.....	19
2.4 Vizualizační prostředí .....	19
2.5 Popis konfigurace reálného PLC.....	20
2.6 Programovací jazyky.....	21
2.6.1 Grafické jazyky.....	22
2.6.2 Textové jazyky .....	22
<b>3 Komerový přízový spinner .....</b>	<b>24</b>
3.1 Popis částí KOPRISu .....	25
<b>4 Technologické uzly linky .....</b>	<b>27</b>
4.1 VN zdroj.....	28
4.2 Podavač příze .....	29
4.3 Vstupní a výstupní zákrut.....	30
4.4 Pumpy .....	32
4.4.1 Peristaltické pumpy .....	32



4.4.2	Šnekové pumpy .....	34
4.5	Dmychadlo .....	34
4.6	Sušící trubice .....	36
4.7	Koncové navíjecí ústrojí .....	37
4.8	Bubnový kolektor .....	38
<b>5</b>	<b>Návrh softwaru pro řízení linky .....</b>	<b>39</b>
5.1	Parametry pro koncové navíjení a rozvádění .....	40
5.2	Řídící struktura experimentální linky .....	41
5.3	Vývojový diagram a struktura řízení .....	42
5.4	Programový blok automatického režimu .....	44
5.4.1	Použité globální proměnné, struktury a datové typy .....	44
5.4.2	Vyčištění tabulky .....	47
5.4.3	Uložení a načtení tabulky .....	47
5.4.4	Přechod mezi stavy automatického režimu .....	48
5.5	Čtení a zápis dat tabulky .....	49
<b>6</b>	<b>Grafické uživatelské rozhraní .....</b>	<b>52</b>
6.1	Auto režim .....	54
<b>7</b>	<b>Návod na použití automatického režimu .....</b>	<b>57</b>
7.1.	Připojení k vizualizaci .....	57
7.2.	Popis automatického režimu .....	58
7.3.	Navigační menu automatického režimu .....	61
	<b>Závěr .....</b>	<b>63</b>
	<b>Seznam použité literatury .....</b>	<b>65</b>

## Seznam obrázků

Obrázek 1: B&R Eggelsberg [1].....	12
Obrázek 2: I/O Systém X67 [4].....	14
Obrázek 3: Systém X20 [3].....	14
Obrázek 4: Cyklus programu PLC.....	15
Obrázek 5: Pracovní prostředí.....	18
Obrázek 6: Vizualizační prostředí.....	19
Obrázek 7: Zobrazení konfigurace projektu.....	20
Obrázek 8: Zobrazení připojeného hardwaru.....	20
Obrázek 9: Schéma řídicího systému.....	21
Obrázek 10: Rozdělení komor KOPRISu [10].....	25
Obrázek 11: 3D model KOPRISu.....	26
Obrázek 12: Transformátor napětí VTS 38 [11].....	28
Obrázek 13: Podavač příze Memminger IRO EFS-800.....	29
Obrázek 14: Vstupní zákrutové zařízení.....	30
Obrázek 15: Výstupní zákrutové zařízení.....	31
Obrázek 16: Krokové motory pump.....	32
Obrázek 17: Řídící jednotky pump.....	33
Obrázek 18: Šnekové pumpy.....	34
Obrázek 19: Dmychadlo Leister.....	35
Obrázek 20: Sušicí trubice.....	36
Obrázek 21: Koncové navíjecí ústrojí.....	37
Obrázek 22: Bubnový kolektor.....	38
Obrázek 23: Diagram řízení technologických uzlů.....	41
Obrázek 24: struktura řízení automatického režimu.....	42
Obrázek 25: Vývojový diagram.....	43
Obrázek 26: Vývojový diagram pro práci s tabulkou.....	50
Obrázek 27: Hlavní strana uživatelského rozhraní.....	53
Obrázek 28: Navíjení a rozvádění v komoře.....	54
Obrázek 29: Automatický režim.....	55
Obrázek 30: Diagram pro práci s tabulkou.....	56
Obrázek 31: VNC Viewer.....	57
Obrázek 32: VNC Viewer overení.....	57
Obrázek 33: Automatický režim - rozbalovací nabídka s akčními členy.....	58
Obrázek 34: Navigační menu automatického režimu.....	61
Obrázek 35: Okno pro načtení a uložení tabulky.....	62

## Seznam tabulek

<i>Tabulka 1: Technické parametry VN zdroje</i> .....	28
<i>Tabulka 2: Technické parametry FT4C52G</i> .....	31
<i>Tabulka 3: Technické parametry krokových motorů pump</i> .....	33
<i>Tabulka 4: Technické parametry dmychadla Leister</i> .....	35
<i>Tabulka 5: Tabulka technologický uzlů</i> .....	39
<i>Tabulka 6: přehled akčních členů z rozbalovací nabídky</i> .....	58
<i>Tabulka 7: Parametry akčních členů</i> .....	60

## Seznam zdrojových kódů

<i>Zdrojový kód 1: Datové typy a struktury</i> .....	44
<i>Zdrojový kód 2: Globální proměnné</i> .....	46
<i>Zdrojový kód 3: Funkce pro vyčištění tabulky</i> .....	47
<i>Zdrojový kód 4: Funkce pro uložení a načtení tabulky</i> .....	48
<i>Zdrojový kód 5: Část programu pro Zákrut 1</i> .....	51

## Seznam zkratk

<b>B&amp;R</b>	Bernecker & Rainer
<b>PLC</b>	programovatelný logický automat
<b>KOPRIS</b>	Komorový Přízový Spinner
<b>VN</b>	Vysoké napětí
<b>POU</b>	Programová organizační jednotka
<b>IP</b>	Stupeň krytí
<b>I/O</b>	Vstup/Výstup
<b>X2X</b>	Komunikační sběrnice

## Úvod

Tématem diplomové práce je navrhnout a vytvořit algoritmus automatického režimu pro řízení experimentální linky, tak aby obsluha mohla snadno měnit konfiguraci dle aktuálního mechanického uspořádání. Experimentální linka slouží k výrobě nanovlákných produktů a nachází se v laboratoři Katedry netkaných textilií a nanovlákných struktur na Technické Univerzitě v Liberci.

Řídící prvky linky jsou od společnosti B&R Automation, je tedy zapotřebí navrhnout algoritmus tak, aby bylo možné vytvořený program implementovat do PLC právě za pomoci softwaru *B&R Automation Studio*. Úvodní část diplomové práce je zaměřená na produkty společnosti B&R Automation, na jejich komunikaci a především na vývojový software *B&R Automation Studio*. Ukážeme si, jak se se studiem pracuje, jaké jsou hlavní programovací jazyky sloužící k vytvoření programu a v neposlední řadě i vizualizační prostředí. Následně se seznámíme s experimentální linkou KOPRIS (komorový přízový spinner) a popíšeme si jeho jednotlivé mechanické a elektrické části. Vysvětlíme si, jaké jednotlivé technické uzly jsou zapotřebí, aby bylo možné vyrobít požadovaný nanovlákný materiál.

Na základě seznámení se s experimentálním ústrojím linky a jednotlivými mechanickými a elektrickými částmi vytvoříme algoritmus tak, aby splnil předpoklady uživatele na konkrétní výrobu nanovlákného materiálu. Vytvořený algoritmus si vysvětlíme a vytvoříme programový blok, který bude sloužit k řízení stroje v automatickém režimu tak, aby obsluha mohla snadno měnit konfiguraci dle aktuálního mechanického uspořádání a zároveň mohla modifikovat parametry produkce. K vytvořenému programovému bloku navrhne grafické uživatelské rozhraní pro snadnou manipulaci s experimentální linkou a s možností ukládání konfigurace a parametrů pro pozdější opakované použití.

Na závěr k vytvořenému softwaru sestavíme jednoduchý návod pro snadnou obsluhu výrobní linky v grafickém uživatelském rozhraní.

# 1 B&R Automation

B&R Automation je společnost, která byla založena roku 1979 v Rakousku a na českém trhu působí od roku 1997. Své hlavní sídlo má v Eggelsbergu, kde v dnešní době provozuje 3 závody. Firma byla založena Erwinem Berneckerem a Josefem Rainerem a v současné době má po celém světě asi 2500 zaměstnanců v 75 zemích. V moderních halách se nachází vše od plně automatických výrobních linek, až po školicí střediska Automation Academy [1].

Firma B&R se během více než 30 let stala jedním z hlavních výrobců v oblasti automatizace. V roce 1980 přišla na trh s čítačem *Unicount* a také programovatelným PLC *Unicontrol*, který byl ve své době moderním řešením automatizace. Dalším důležitým datem v historii firmy je rok 1993, kdy firma představila jeden ze svých prvních průmyslových počítačů (IPC2000). V letech 1999 a 2000 se firma podílela na vývoji komunikačního standardu *Ethernet Powerlink*, který od té doby využívá ve svých produktech. V tomto období také vyvinula a začala nabízet software *Automation Studio*, který v dnešní době již dosáhl 4. generace [1].



Obrázek 1: B&R Eggelsberg [1]

## 1.1 Produkty B&R

B&R zajišťuje kompletní řešení pro průmyslovou automatizaci, nabízí vše potřebné pro automatizaci strojů, od jednoduchých až po velmi rozsáhlé systémy. Produkty B&R se zakládají na hladké spolupráci mezi sebou. Firma nabízí k programování, konfiguraci a vizualizaci všech svých produktů software *Automation Studio*.

### 1.1.1 Programovatelné logické automaty (Systém X20, X67)

PLC (*Programmable Logic Controller*) neboli programovatelný logický automat či pouze programovatelný automat je uživatelsky programovatelný řídicí systém přizpůsobený pro řízení technologických a průmyslových procesů. V dnešní době programovatelné logické automaty zasahují téměř do všech oblastí průmyslu [2].

PLC jsou základním kamenem pro řízení automatizovaných linek, robotů, dopravy a mnoha dalších oblastí. Značné využití mají i mimo průmysl. Lze je využít pro realizaci inteligentního domu, klimatizačního zařízení, dále pro ovládání garážových vrat, mycích linek, balicích strojů, prodejních automatů apod. [2].

Mezi základní výhody použití řízení pomocí PLC například patří:

- Rychlá realizace
- Spolehlivost, odolnost a diagnostika
- Nekončící změny v zadávání

Společnost B&R nabízí programovatelné automaty ve dvou provedeních, jde o řady typu X20 a X67. Názvy odpovídají stupni krytí, který daný systém nabízí – IP20 a IP67.

Systém X20 umožňuje kombinovat komponenty v závislosti na požadavcích uživatele. Je založen na bezproblémové integraci, vysokém výkonu a flexibilitě. Mechanické rozdělení s sebou přináší mnoho výhod. Jednou z nich je efektivnější využití prostoru pro dosažení vyšší „hustoty“ komponent, dále také umožňuje vyměňování elektroniky za chodu systému a do svorek lze zapojit až tři vodiče, tudíž nejsou potřeba další propojovací svorky [3].

Systemy X67 (Obrázek 2) jsou konstruovány pro nasazení v náročných podmínkách. Systemy nabízí kompatibilitu s prvky X20 [4].



*Obrázek 2: I/O Systém X67 [4]*

System X20 (Obrázek 3), lze z konstrukčního provedení rozdělit do tří základních rozebíratelných částí:

- Svorka
- Modul s elektronikou
- Sběrný modul



*Obrázek 3: Systém X20 [3]*

## 1.1.2 Vykonávání programu PLC

Program PLC je tvořen posloupností instrukcí a příkazů jazyka. Základním znakem je jeho cyklické vykonávání v programové smyčce, a proto se programátor nemusí starat o to, aby po konci programu vrátil jeho vykonávání opět na začátek. Naopak při dlouhodobém setrvání programu v programové smyčce může dojít k překročení doby cyklu a následnému selhání [2].

Po každém vykonání poslední instrukce uživatelského programu je předáno řízení systémovému programu, který zajistí tzv. otočku cyklu. V ní jsou nejprve na výstupní periférie vyslány aktuálně vyčíslené hodnoty obrazů výstupů, dále jsou provedeny režijní operace systému, které zahrnují aktualizaci časových údajů pro časovače a systémové registry, ošetření komunikace, diagnostiku a řadu dalších režijních úkonů. Na závěr jsou načteny aktuální hodnoty fyzických vstupů, které jsou po dobu následujícího cyklu uloženy do paměťových obrazů. Poté je opět předáno řízení první instrukci uživatelského programu [2].



Obrázek 4: Cyklus programu PLC



## 1.2 Komunikační standardy

### 1.2.1 Powerlink

B&R využívá komunikační protokol *Powerlink*, který byl uveden na trh v roce 2001 firmou B&R. *Powerlink* je systém vycházející ze standardu *Ethernet* a využívá komunikační model producent/konzument, což vede k větší propustnosti sítě a lepšímu výkonu. Tím že *Powerlink* vychází ze standardu *Ethernet*, tak nepotřebuje žádný dodatečný hardware a lze pro něj využít všechny čipy, systémy vytvořené pro *Ethernet*. *Powerlink* je ideálním protokolem pro pohony, vstupy a výstupy, vizualizaci a výměnu dat mezi systémy automatů [1] [5] [6].

### 1.2.2 X2X Link

Všechny moduly X20 jsou mezi sebou spojeny sběrnici X2X Link. Standard X2X Link spojuje moduly vedle sebe, nebo umožňuje spojení dvou modulů vzdálených několik metrů od sebe. Každý modul má vlastní přijímač X2X a vysílač X2X. Všechna data, která dorazí na přijímač, jsou okamžitě předána na vysílač [6].

X2X protokol definuje pevnou délku cyklu, ve kterém se opětovně odesílají čtyři základní části (*SyncOut*, *SyncIn*, *AsynOut*, *AsynIn*). Nejrychlejší možný časový cyklus je 100  $\mu$ s. Časové cykly se konstantně opakují [6].

### 1.2.3 Profibus, DeviceNet, Can

*Profibus* je jednoduchý vysokorychlostní protokol pro klasické automatizační úlohy v oblasti výrobního inženýrství. Hlavním rysem protokolu *Profibus* je poskytování extrémně bezpečných procesů [1] [7].

*DeviceNet* je otevřený standard průmyslové sběrnice. Specifikace protokolu a údržba standardu *DeviceNet* jsou pod dohledem nezávislé a otevřené uživatelské organizace ODVA (*Open DeviceNet Vendor Association*) [1] [7].

*Can* (*Controller area network*) sběrnice je hojně používaná pro vybrané B&R I/O systémy, které fungují přes speciální protokol na CAN sběrnici [7].

## 2 Automation studio

*Automation Studio* je programové prostředí vyvinuté pro produkty B&R, jako jsou například regulátory, řídicí jednotky nebo vizualizační jednotky. Uživatel si může vybrat z mnoha programovacích jazyků, diagnostických nástrojů a editorů k vytvoření svých projektů.

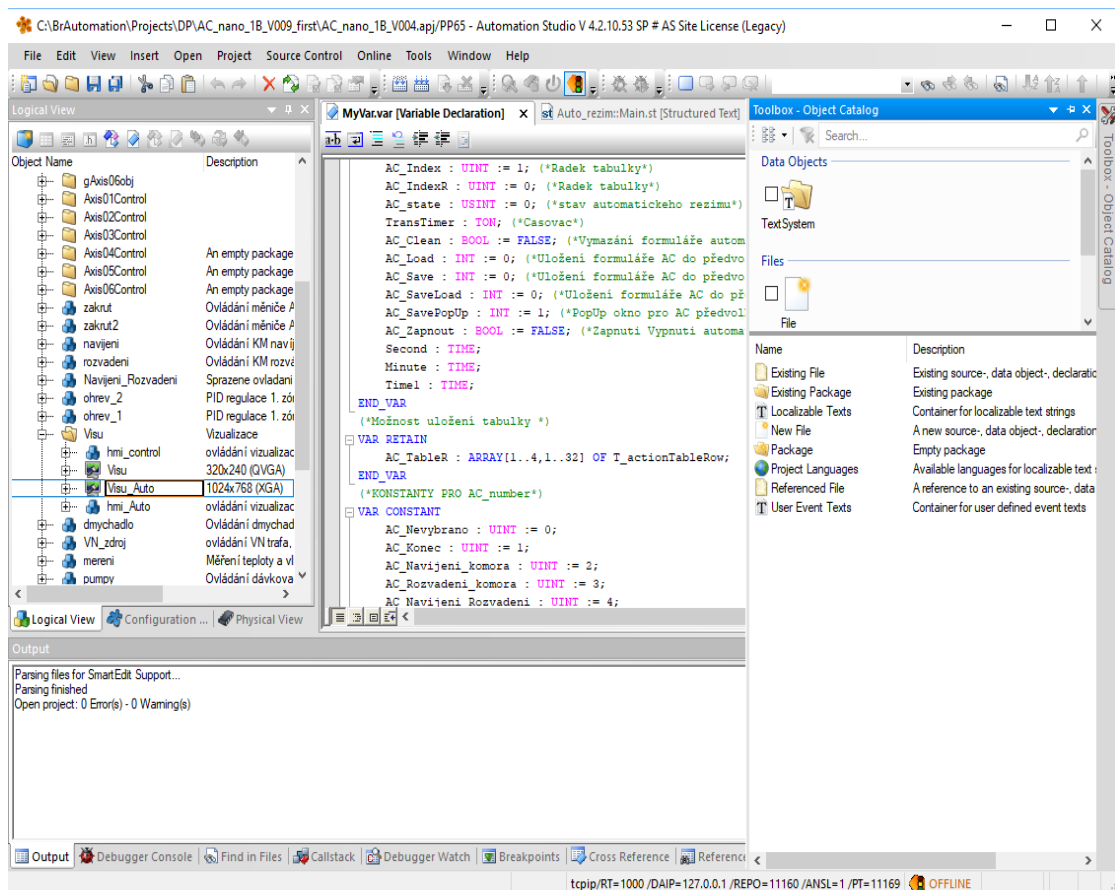
### 2.1 Automation Runtime

Součástí prostředí *Automation Studio* je operační systém reálného času *Automation Runtime*, který tvoří softwarovou platformu všem produktům společnosti B&R. *Automation Runtime* zajišťuje chod programů a poskytuje konfiguraci cílových systémů. Další funkcí je možnost využití osmi různých cyklických tříd.

### 2.2 Vytvoření programu a pracovní prostředí

*Automation Studio* je rozděleno do několika částí, každá z nich má své specifické funkce a úlohy. Pracovní prostředí (Obrázek 5) a jeho jednotlivé části jsou následující:

- Menu a pracovní lišta poskytují přístup k rozsáhlým funkcím.
- V levé části pracovního prostředí se nachází organizátor projektu, který obsahuje tři základní záložky, kterými se vytváří nebo editují komponenty projektu (*Logical View*, *Configuration View* a *Physical View*).
- V pravé části pracovního prostředí se nachází Objektový katalog (*Toolbox*), umožňuje snadnou implementaci nástrojů, funkcí a objektů.
- Uprostřed se nachází hlavní okno projektu, kde lze programovat a konfigurovat data projektu.
- Dolní část pracovního prostředí obsahuje výstupní okno, které zobrazuje informace při kompilaci projektu



Obrázek 5: Pracovní prostředí

Každý vytvořený projekt obsahuje globální proměnné (*Global.var*), které jsou využívány pro celý projekt. Jednotlivé programy obsahují navíc lokální proměnné, které jsou deklarovány a využívány jen daným programem. Lze také vytvořit vlastní datové typy (např. struktury), které se dají dále vnořovat. Stejně jako proměnné, tak i struktury můžeme rozdělit na globální (*Global.typ*) a lokální (*Types.typ*).

Nedílnou součástí každého projektu jsou také knihovny. Lze vytvářet i vlastní knihovny a poté je využívat ve svých projektech anebo si stáhnout již vytvořené knihovny.

Záložka *Configuration View* umožňuje softwarovou správu použitého hardwaru. V této záložce lze vytvořeným programům přiřadit jednu z osmi cyklických tříd, které se liší rychlostí vykonání cyklu. Zároveň je nutné si uvědomit, že rychlejší cyklické třídy jsou náročnější na procesor. Záložka *Logical View* slouží ke správě programové části projektu a je nezávislá na hardwaru.

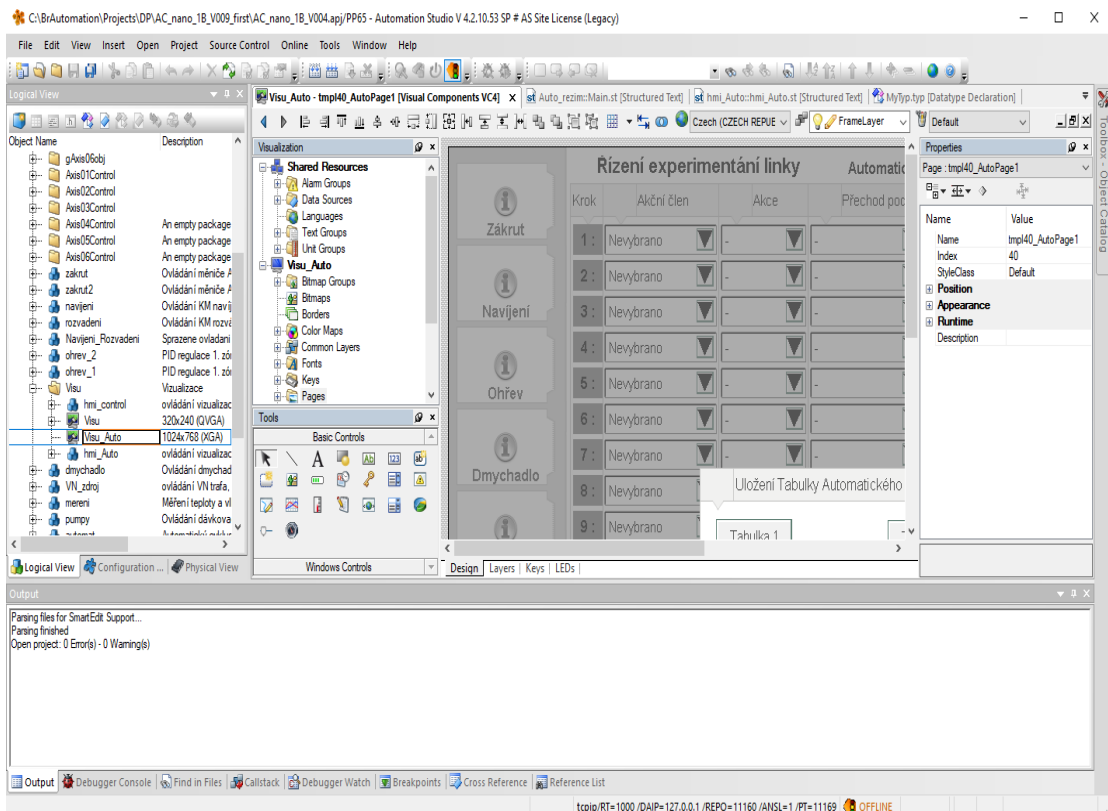
## 2.3 Práce se simulátorem

Práce se simulátorem je nedílnou součástí každého projektu. Simulace umožňují simulovat konfigurace, jako kdyby byly na skutečném PLC procesoru. Simulace jsou převážně využívány v rané části projektu, kdy eliminují složité hardwarové zapojení a zároveň umožňují rychlé testování projektu.

Pro přesun projektu do simulátoru se používá tlačítko *Transfer*, které se nachází v horní liště. Při práci se simulátorem se simulace sleduje pomocí diagnostického nástroje *Watch*.

## 2.4 Vizualizační prostředí

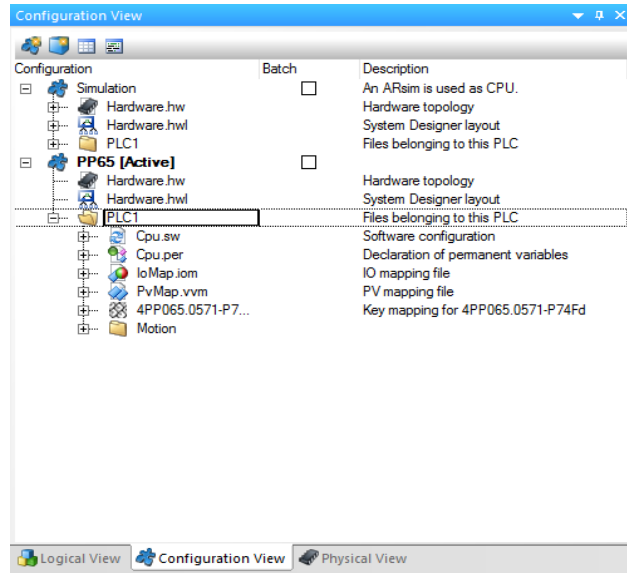
Vizualizace slouží k usnadnění ovládání a pomáhá zobrazit složité děje. Jedná se o grafické zobrazení procesu, které umožňuje ovládat, nastavovat a sledovat vybrané hodnoty pomocí vizualizační obrazovky. Vizualizační prostředí je součástí softwaru *Automation Studio Visual Components* (VC). To umožňuje tvorbu a spuštění vizualizace společně s hlavní částí projektu. Pracovní prostředí pro tvorbu vizualizace je zobrazeno na následujícím obrázku (Obrázek 6).



Obrázek 6: Vizualizační prostředí

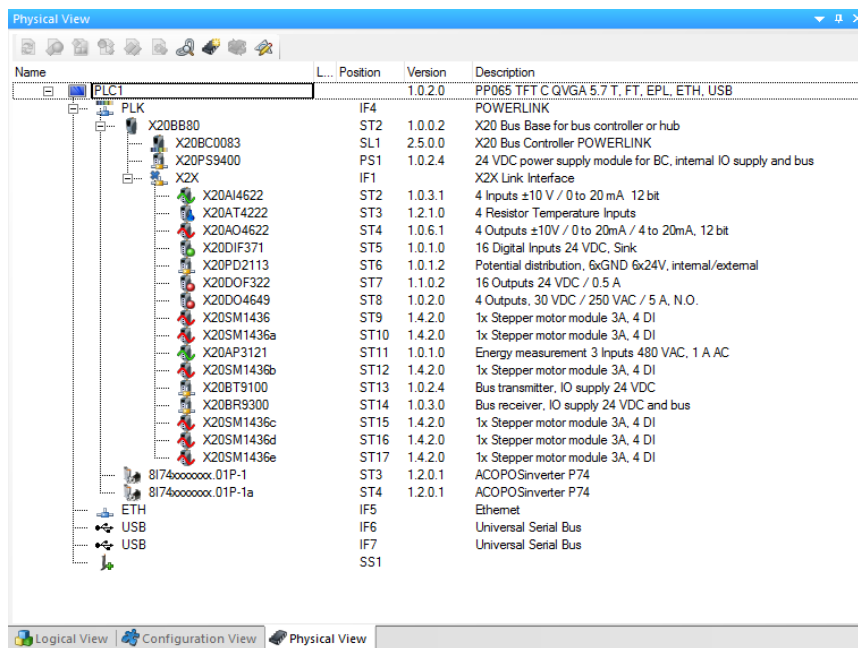
## 2.5 Popis konfigurace reálného PLC

V záložce *Configuration View* se řídí hlavní konfigurace projektu. Konfigurace rozhoduje o tom, jaký hardware a software je právě aktivní.



Obrázek 7: Zobrazení konfigurace projektu

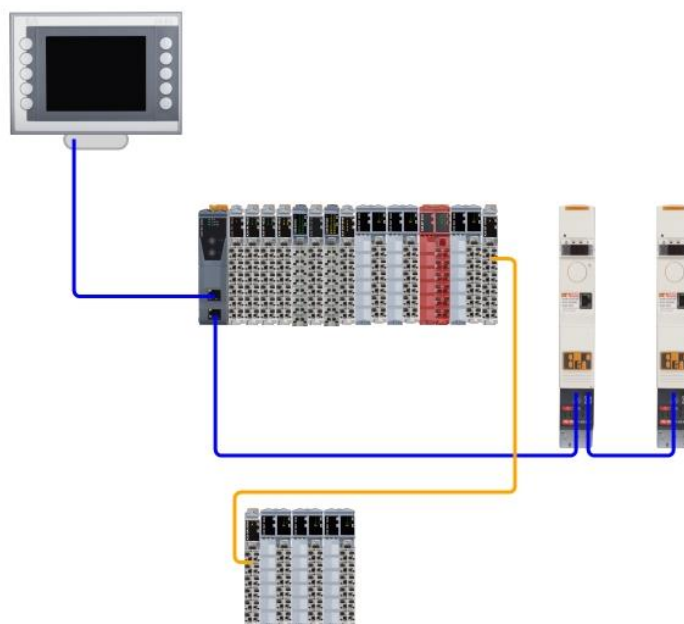
Hardwarové části systému se konfiguruji a ovládají v záložce *Physical View*. Do *Physical View* se jednotlivé moduly přiřazují liniově, tak jak jsou zapojeny hardwarově vedle sebe komunikací X2X. Jediné proměnné se přiřazují na moduly pomocí „I/O mapping“.



Obrázek 8: Zobrazení připojeného hardwaru

Na předchozím obrázku (Obrázek 8) je vidět struktura zapojení hardwarové části stroje. Řídicím systémem je programovatelný logický automat, který je zapojen v kombinaci s grafickým dotykovým panelem. Řídicí jednotka sběrnice propojuje integrovaný rozbočovač X2X link I/O s komunikačním protokolem POWERLINK.

- Pomocí rozbočovače X2X jsou připojeny jednotlivé moduly digitálních/analogových vstupů a výstupů technologických uzlů a také moduly krokových motorů.
- Pomocí Ethernetového kabelu a protokolu POWERLINK jsou připojeny dva frekvenční měniče.



Obrázek 9: Schéma řídicího systému

## 2.6 Programovací jazyky

Původně každý z výrobců PLC nabízel své systémy s vlastními variantami programovacích jazyků, a to postupem času přineslo značné komplikace. Tuto nepřehlednost následně vyřešila mezinárodní norma IEC 61131-3, která sjednocuje programovací jazyky pro PLC. *Automation Studio* nabízí mnoho programovacích jazyků: *B&R Automation Basic*, *Instruction List*, *Ladder Diagram*, *Function Block Diagram*, *Structured Text*, *SFC*, *CFC*, *ANSI C*, *ANSCI C++*.

## 2.6.1 Grafické jazyky

### a) *Ladder Diagram*

*Ladder Diagram* (LD) je jedním ze základních grafických jazyků využívaných v programu *Automation Studio*. LD je založen na grafické interpretaci reléové logiky a je podobný liniovým schémátům v elektrotechnice. Mezi dvěma svislými čarami (které se nazývají levá a pravá napájecí sběrnice) se nachází síť kontaktů, cívek, funkcí a funkčních bloků. Tato sérioparalelní síť je zapojena zleva doprava. LD se využívá při programování jednodušších logických úloh [8] [9].

### b) *Function Block Diagram*

Jazyk funkčních bloků (FBD) je dalším grafickým jazykem využívaným v programu *Automation Studio*. FBD je podobný LD, ale na rozdíl od něj využívá na místo reléové logiky vzájemné propojení obdélníkových značek výrokové logicky, čítačů, časovačů atd. Jednou z největších výhod FBD je online sledování stavů [8] [9].

## 2.6.2 Textové jazyky

### a) *Structured Text*

Strukturovaný text (ST) je jeden z nejpoužívanějších vyšších textových programovacích jazyků v programu *Automation Studio*. Svou strukturou a příkazy je velice podobný Pascalu a ANSI C. ST obsahuje klasické prvky vyšších programovacích jazyků, jako jsou např. příkazy větvení (*IF*, *CASE OF*), nebo smyčky (*FOR*, *WHILE*). Všechny příkazy jsou odděleny středníkem. ST se využívá v programech, kde se složitější operace (např. práce s databázemi) naprogramují mnohonásobně rychleji, než u grafických jazyků. ST je vhodný pro práci s daty a řetězci. Velký nárok je kladen na znalost všech příkazů, syntaxí a na schopnosti programátora. ST není vhodný při zpracování velkého počtu logických vstupů a výstupů [8] [9].

## ***b) ANSI C***

V programu *Automation Studio* lze také programovat v *ANSI C* a *ANSI C++*. *ANSI C* je jeden z nejrozšířenějších programovacích textových jazyků pro psaní systémového softwaru ale i pro psaní aplikací. Tento programovací jazyk začíná využívat čím dál víc programátorů pro programování PLC. Programování v *ANSI C* se doporučuje spíše pokročilým programátorům, jelikož jeho syntaxe je složitější [8] [9].



### 3 Komorový přízový spinner

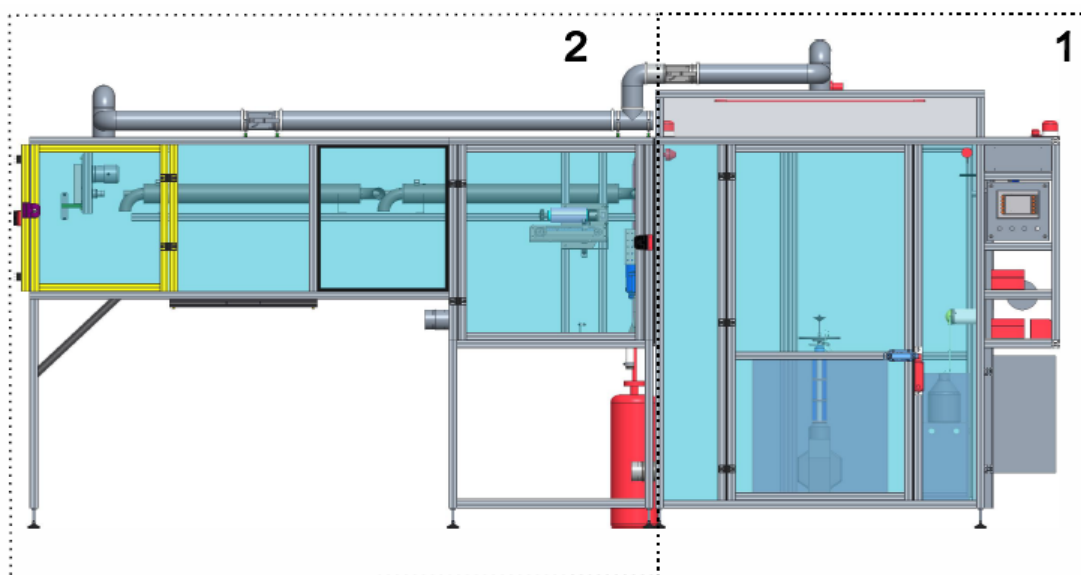
AC zvlákňovací zařízení s označením **KOPRIS 2** (*komorový přízový spinner, verze 2*) slouží k výrobě jádrové příze s obsahem nanovláken vyrobených metodou zvlákňování polymerního roztoku v elektrickém poli s vysokou intenzitou. Pole je vytvářeno na zvlákňovací elektrodě zdrojem regulovaného střídavého vysokého napětí 0-35 kV. Zařízení se skládá ze dvou základních částí: komory 1 a komory 2 (viz Obrázek 10).

V komoře 1 dochází ke zvlákňování polymerního roztoku a nanášení vyrobených nanovláken na pohybující se jádrovou přízi, z tohoto důsledku se komora nazývá zvlákňovací.

V komoře 2 dochází k sušení a fixaci nanoseného nanovláknenného pláště, dodatečnému přikroucení materiálu a jeho navinutí na výstupní cívku. Komora se nazývá sušicí a je u stroje KOPRIS 2 zcela uzavřena, což umožňuje práci bez kontaminace vnějšího prostředí [10].

*Komorový přízový spinner* je experimentální linkou, zejména z důvodu možnosti změny mechanického uspořádání technologických uzlů. Například změnou dopravy polymerního roztoku ke zvlákňovacím elektrodám. Dopravovat roztok je možné například jenom pomocí jedné peristaltické pumpy místo dvou. V jiném případě lze použít místo peristaltických pump pumpy šnekové, v kombinaci s kádinkou a zvlákňovací elektrodou. Mezi další možnosti výroby jádrové příze s obsahem nanovláken, je možnost použití jen vstupního zákrutu a například jen jedné topné trubice.

Další možností nanášení nanovláknena je použití posuvného bubnového kolektoru, kde dochází k nanášení nanovláken na válcovou plochu rotujícího bubnu. Použitím tohoto rotujícího bubnového kolektoru, který lze instalovat do zvlákňovací komory, je možné vyrábět plošný nanovláknenný materiál.



Obrázek 10: Rozdělení komor KOPRISu [10]

### 3.1 Popis částí KOPRISu

Následující text popisuje následující obrázek (Obrázek 11), který odpovídá 3D modelu *komorového přízového spinneru* a popisuje nejdůležitější části stroje.

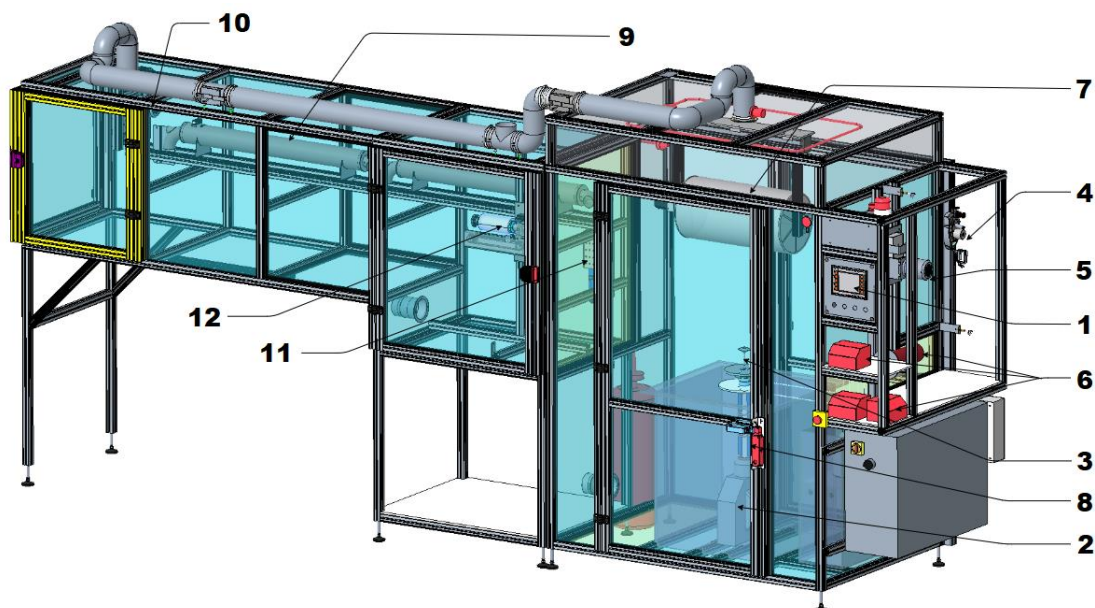
Ovládání všech procesů zajišťuje obsluha pomocí dotykového ovládacího panelu. Pro dosažení potřebného vysokého napětí pro zvlákňovací proces je využito transformátoru. Elektrická ochrana přívodu VN z transformátoru do vnitřního prostoru zvlákňovací komory je zajištěna soustavou ochranných odporů. Vývod VN je v komoře připevněn ke zvlákňovací elektrodě.

Vnitřní prostor umožňuje instalaci různých typů elektrod a také vyššího počtu elektrod. Při výrobě jádrové příze s obsahem nanovláken je nosné jádro odvíjeno ze zásobní cívky a podáváno podavačem do zákrutového zařízení. Průchodem nosného materiálu zákrutovým zařízením je dosaženo specifického prostorového pohybu jádra, v důsledku čehož je nanovlákný materiál stoupající volně z elektrody vzhůru zachycen na nosném jádru. Polymerní roztok je ke zvlákňovacím elektrodám dopravován např. pomocí peristaltických pump, jejichž funkční část je umístěna uvnitř zvlákňovací komory a ovládací část je umístěna vně komory.

Nosné jádro s ovinutou nanovláknennou vrstvou následně vstupuje do sušící a fixační komory s prouděním předehřátého vzduchu z horkovzdušného dmyhadla. V případě potřeby je možné přízový materiál dále přikroutit pomocí příkrucovacího

zařízení. Výsledný materiál je následně navinut na výstupní cívku pomocí navíjecího zařízení.

V případě výroby plošného nanovláknenného materiálu je do prostoru zvláknovací komory připevněn bubnový kolektor, který rotuje a zároveň se periodicky posouvá. V případě, kdy není bubnový kolektor využíván, je bezpečně umístěn k rámu komory sušící.



Obrázek 11: 3D model KOPRISu

#### Legenda

1. Ovládací panel
2. Transformátor (VTS 38 KPB Intra)
3. Zvláknovací elektroda (různé typy)
4. Podavač jádrového materiálu
5. Zákrutové zařízení
6. Peristaltické pumpy, (Syringe pump NE-9000)
7. Rotující bubnový kolektor
8. Ochranné odpory přívodu VN do pracovní části komory
9. Sušící/fixační komora (do 150 °C)
10. Přikrucovací zařízení
11. Zdroj předehřátého vzduchu (do 150°C)
12. Koncové navíjecí zařízení

## 4 Technologické uzly linky

Z hlediska technologie výroby nanovláknenných produktů se jedná o experimentální zařízení. Tato skutečnost v praxi znamená, že postupem času jsou jednotlivé pracovní uzly stroje zdokonalovány (např. zákrutová ústrojí), inovovány či vyměněny za jiné (komerčně vyráběný podavač příze), případně se mění jejich počet podle požadavků konkrétního experimentu (např. počty šnekových pump jsou často různé).

Z hlediska způsobu použití celého stroje můžeme rozlišit dva případy:

- 1) *Experimentální chod.* Ovládání jednotlivých technologických uzlů a nastavení jejich parametrů je zcela v kompetenci obsluhy. Posloupnost zapínání a vypínání jednotlivých uzlů není předem dána a odvíjí se od okamžité situace na stroji.
- 2) *Poloprovozní chod.* Zde je cílem vyrobit větší množství určitého nanovláknenného produktu, což typicky znamená, že stroj je v chodu několik desítek minut až jednotky hodin. Během této doby může docházet k zastavování a opětovnému spouštění produkce. Protože požadovaný nanovláknenný produkt je již znám a jsou určeny i jeho cílové vlastnosti, je žádoucí, aby celý chod stroje byl pokud možno co nejméně „negativně“ ovlivněn nedisciplinovaností obsluhy. Jednotlivé technologické uzly se tedy musí zapínat a vypínat dle předem definované posloupnosti a tyto procesy je tedy vhodné v co největší míře automatizovat.

Námi navrhovaný software by tuto problematiku měl řešit.

Aby byla možná výroba produktů s nanovláknennou vrstvou, je zapotřebí hned několika technologických mechanismů (uzlů), které správnou kombinací zapříčiní dosažení požadovaného produktu. Tyto „uzly“ již byly popsány v popisu KOPRISu v kapitole 3.1. V této kapitole si popíšeme ty nejdůležitější a stěžejní pro uživatele, který bude linku ovládat.

## 4.1 VN zdroj

K vytvoření vysokého napětí pro zvlákňovací proces slouží transformátor typu VTS 38 od výrobce KPB Intra (viz Obrázek 12). VTS 38 je jednofázově jednopólově izolovaný transformátor, určený pro použití v sítích vysokého napětí. K napájení lze použít regulační autotransformátor, nebo lze připojit libovolný externí zdroj střídavého napětí. Prostor transformátoru je od zvlákňovací komory oddělen samostatnou komorou kvůli zamezení znečištění. Vývod transformátoru je připevněn ke zvlákňovací elektrodě a používá se jak pro délkový tak pro plošný produkt. Ovládání transformátoru je řešeno pomocí čtyř stykačů. Technické parametry zdroje najdeme v tabulce (Tabulka 1).



Obrázek 12: Transformátor napětí VTS 38 [11]

Tabulka 1: Technické parametry VN zdroje

Atribut	Hodnota [jednotka]
Nejvyšší napětí soustavy:	36/38.5 [kV]
Zkušební napětí střídavé:	70/80 [kV]
Zkušební napětí impulsní:	170/180 [kV]
Jmenovité primární napětí:	$3000/\sqrt{3} - 35000/\sqrt{3}$ [V]
Jmenovité sekundární napětí:	100/ $\sqrt{3}$ , 110/ $\sqrt{3}$ , 120/ $\sqrt{3}$ [V]
Jmenovité pomocné napětí:	100/3, 110/3, 120/3 [V]
Třída přesnosti - měření:	0.2, 0.5, 1, 3
Třída přesnosti - jištění:	3P, 6P
Jmenovitá zátěž:	10, 30, 50, 75, 100, 150 [VA]
Krajní výkon:	500 [VA]
Jmenovitý kmitočet:	50 [Hz]
Hmotnost:	33 [kg]
Teplotní třída izolace:	E
Provozní podmínky:	provozní teplota od -5 do +40°C

## 4.2 Podavač příze

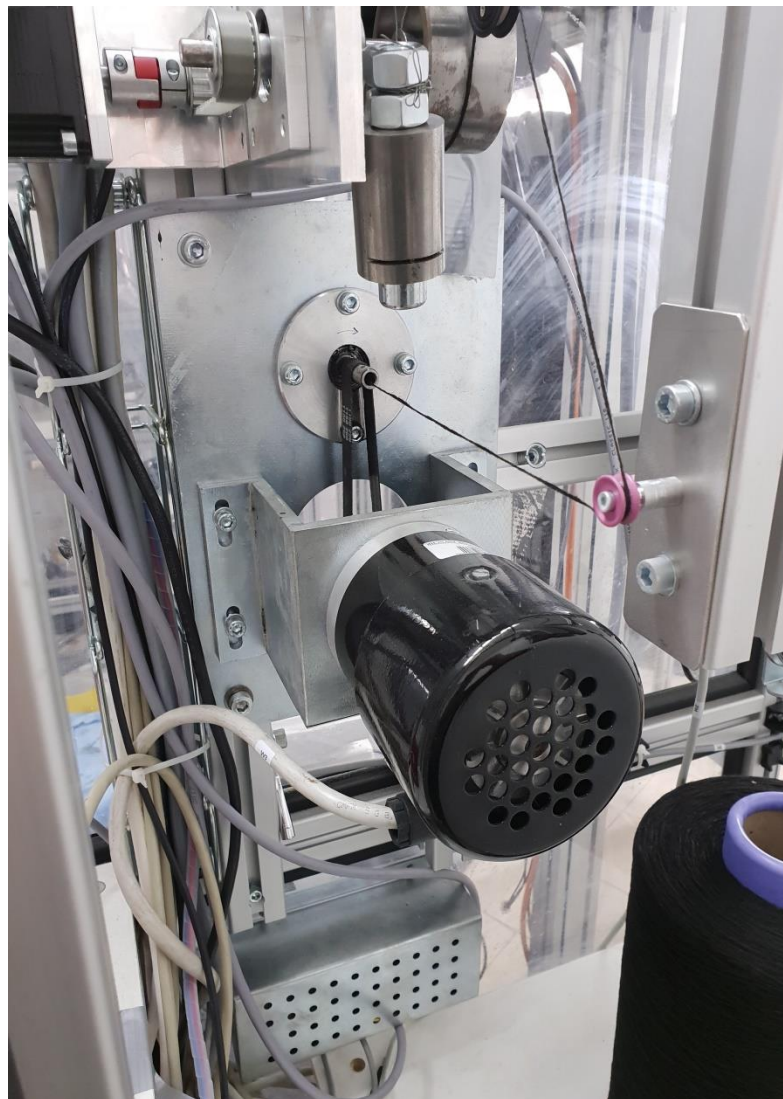
Podavač příze je použit při výrobě délkového produktu a umožňuje zajištění konstantního napětí v přízi. Podavač disponuje regulací tahu, který pomocí snímače měří napětí příze a nastavuje tak rychlost posuvu. Podavač je konstruovaný jako kompaktní zařízení od výrobce Memminger IRO EFS-800 a nachází se vně zvláknovací komory.



*Obrázek 13: Podavač příze Memminger IRO EFS-800*

### 4.3 Vstupní a výstupní zákrut

Příze, před vstupem do zvláknovací komory (viz Obrázek 14) a výstupem ze sušicí trubice (viz Obrázek 15), prochází zákrutovým zařízením, kde dochází k tzv. zákrutu. Zákrut slouží k dosažení prostorového pohybu jádra, v důsledku čehož je nanovláknenný materiál zachycen na nosném jádru. Zákrutové zařízení je poháněno asynchronním trojfázovým elektromotorem od výrobce ATAS Náchod (typ FT4C52G) o výkonu 500W. Napájení a regulace otáček je zajištěna frekvenčním měničem od výrobce B&R Automatisation (typ 8I74S200055.01P-1). Zákrutové mechanismy se používají k výrobě jádrové příze s obsahem nanovláken. Technické parametry elektromotoru najdeme v tabulce (Tabulka 2).



Obrázek 14: Vstupní zákrutové zařízení



Obrázek 15: Výstupní zákrutové zařízení

Tabulka 2: Technické parametry FT4C52G

Atribut	Hodnota [jednotka]
Napětí:	3 x 230 [V ]
Kmitočet:	400 [Hz]
Výkon:	500 [W]
Proud:	3 [A]
Otáčky:	23700 [ot/min]
Krytí:	IP 44
Hmotnost:	2,67 [Kg]



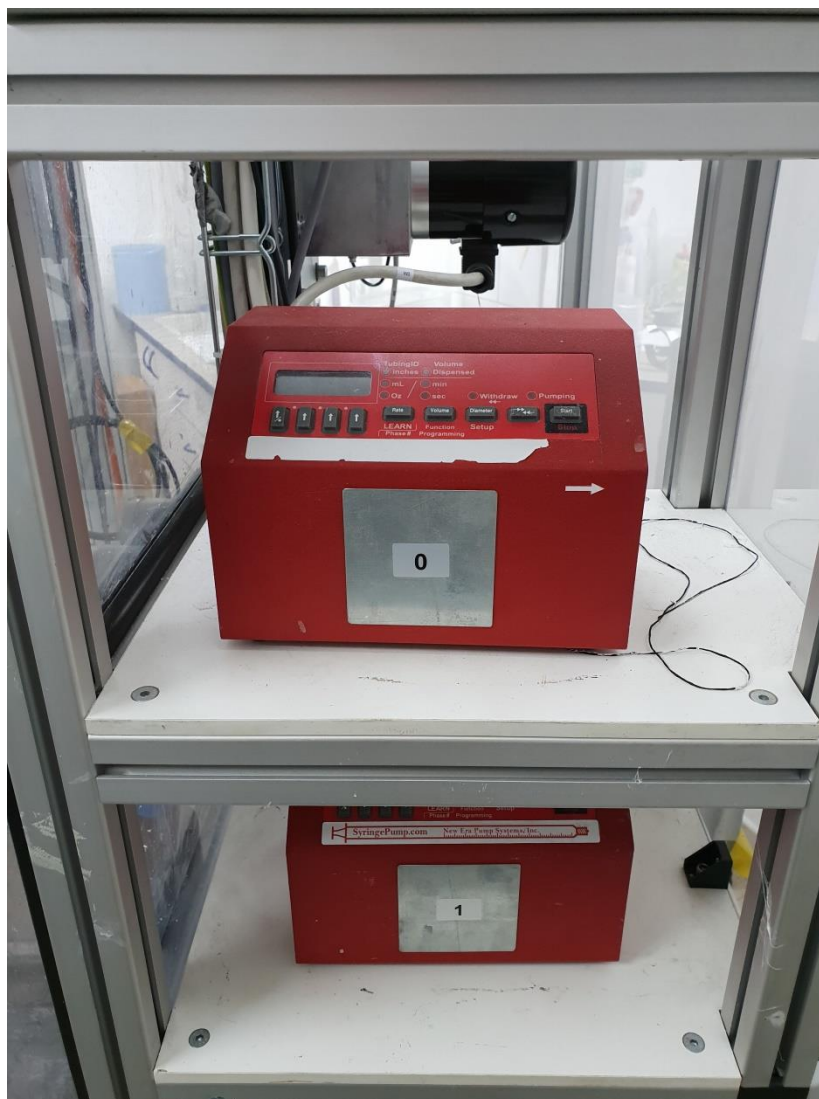
## 4.4 Pumpy

### 4.4.1 Peristaltické pumpy

Pomocí dvou peristaltických pumpje zajištěna doprava a přesné dávkování polymerního roztoku ke zvláknovacím elektrodám. Pohon pump je zajištěn krokovými elektromotory typu B&R Automatisation (typ 80MPF3.250S000-01), které jsou napájeny a řízeny pomocí programovatelných jednotek typu NEPS NE-9000. Ovládací část je umístěna vně komory (viz Obrázek 17). Obě pumpy jsou napájeny 24V a nacházejí se ve zvláknovací komoře (viz Obrázek 16). Pumpy se využívají jak pro výrobu délkového, tak plošného produktu. Technické parametry krokových elektromotorů najdeme v tabulce (Tabulka 3).



Obrázek 16: Krokové motory pump



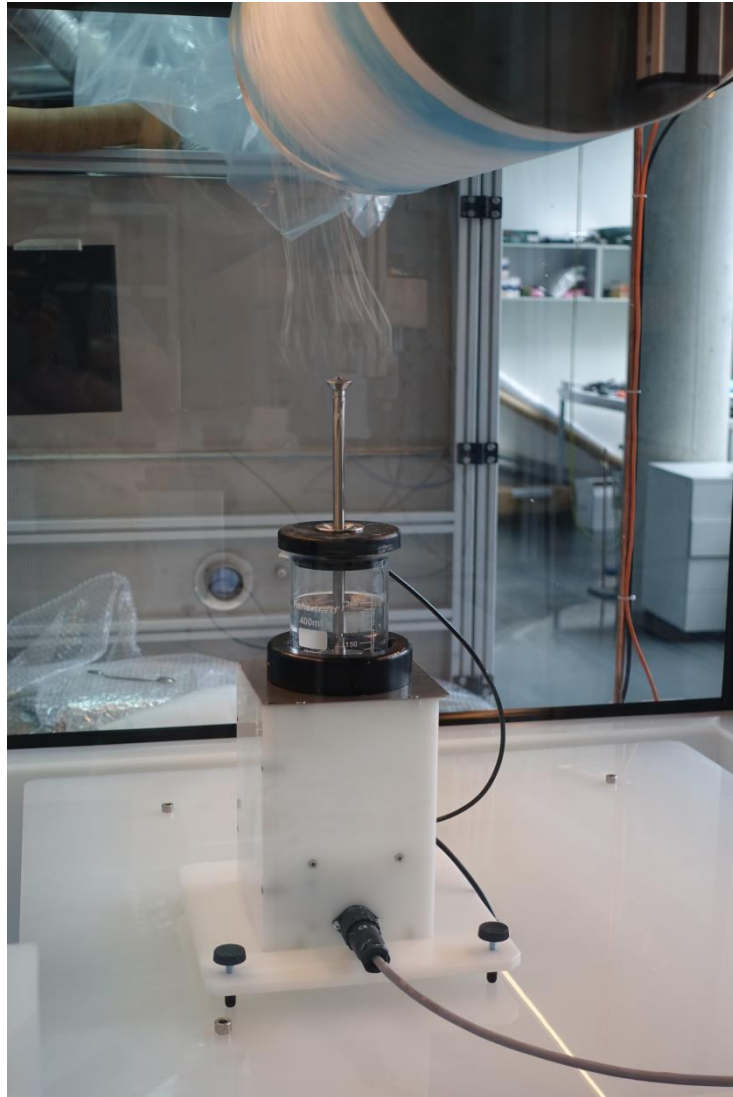
Obrázek 17: Řídící jednotky pump

Tabulka 3: Technické parametry krokových motorů pump

Atribut	Hodnota [jednotka]
Proud v sériovém zapojení:	2,5 [A]
Proud v paralelním zapojení:	5 [A]
Odpor v sériovém zapojení:	1,52 [ $\Omega$ ]
Odpor v paralelním zapojení:	0,38 [ $\Omega$ ]
Indukčnost v sériovém zapojení:	5,6 [mH]
Indukčnost v paralelním zapojení:	1,4 [mH]
Startovní moment:	1,2 [Nm]
Přidržovací moment:	1,7 [Nm]
Bezproudivý moment:	< 45 [mNm]
Moment setrvačnosti:	440 [gcm <sup>2</sup> ]

#### 4.4.2 Šnekové pumpy

Rotační šneková pumpa je umístěna ve zvláknovací komoře nad VN transformátorem společně se zvláknovací elektrodou a kádinkou s roztokem (viz Obrázek 18). Šneková pumpa je poháněna pomocí krokového elektromotoru od firmy B&R Automatisation (typ 80MPF3.250S000-01) a je řízena pomocí modulu krokového motoru. Technické parametry krokových elektromotorů najdeme v předchozí tabulce (Tabulka 3).



Obrázek 18: Šnekové pumpy

#### 4.5 Dmychadlo

Dosažení požadované teploty přehřátého vzduch, který vstupuje do sušicí trubice, zajišťuje horkovzdušné dmychadlo od výrobce Leister typu MISTRAL 6 SYSTÉM (viz Obrázek 19), které je umístěno v sušicí komoře. Dmychadlo je schopno

nepřetržitého provozu při požadované teplotě a průtoku vzduchu. Regulace teploty je zajištěna pomocí PID regulátoru v PLC. Výkon průtok dmyhadla je ovládán pomocí napěťových analogových signálů na modulu. Technické parametry dmyhadla najdeme v tabulce (Tabulka 4).



*Obrázek 19: Dmyhadlo Leister*

*Tabulka 4: Technické parametry dmyhadla Leister*

<b>Atribut</b>	<b>Hodnota [jednotka]</b>
Napětí:	230 [V]
Příkon:	4500 [W]
Maximální teplota:	650 [°C]
Průtok vzduchu:	100 – 400 [l/min]
Statický tlak:	3000 [Pa]
Úroveň hlučnosti:	65 [dB]
Hmotnost:	1,5 [kg]
Frekvence:	50/60 [Hz]

## 4.6 Sušicí trubice

Sušicí trubice slouží k udržení požadované teploty příze pro fixaci nanovláknenné vrstvy. Příze prochází soustavou dvou trubic, ve kterých jsou 1kW topná tělesa. Sušicí trubice jsou umístěny za zvlákňovací komorou (viz Obrázek 20). Regulace teploty je nezávislá a je zajištěna pomocí PID regulátorů implementovaných v PLC.



*Obrázek 20: Sušicí trubice*

## 4.7 Koncové navíjecí ústrojí

Navíjecí ústrojí slouží ke koncovému soukání příze na cívku, mechanismus je vyobrazen na následujícím obrázku (Obrázek 21). K navíjení příze jsou použity dva elektronicky řízené krokové motory. Cívka je upevněna v rámu a je poháněna prvním krokovým motorem, který vykonává rotační pohyb. Druhý motor pohání přes kladku řemínek, na kterém je připevněna rozváděcí kladka, která koná přímočarý vratný pohyb podle programu. Sloučením obou pohybů je docíleno tvorby křížového návínů příze na cívku.



*Obrázek 21: Koncové navíjecí ústrojí*

## 4.8 Bubnový kolektor

Bubnový kolektor je použit při výrobě plošného produktu. Aby bylo možné nanášet nanovláknenný materiál po celé šířce bubnu, může buben kromě rotačního pohybu vykonávat také pohyb axiální. Rotační pohyb a axiální pohyb (tzv. rozvádění) zajišťují dva krokové elektromotory, které jsou připojeny na moduly krokového motoru společně s indukčním koncovým snímačem polohy.



*Obrázek 22: Bubnový kolektor*

## 5 Návrh softwaru pro řízení linky

Cílem práce bylo na základě zjištěných skutečností navrhnout a vytvořit nový software pro řízení stroje v automatickém režimu, tak aby obsluha mohla snadno měnit konfiguraci dle aktuálního mechanického uspořádání. Původní software pro ovládání linky v manuálním režimu zůstane zachován, obsluha tak stále může používat stroj dle dosavadních zvyklostí.

Mechanické uspořádání linky je popsáno v kapitole 3.1. Aby bylo možné navrhnout algoritmus, bylo zapotřebí kromě mechanického uspořádání se seznámit i se stávajícím programovým řešením POU (programová organizační jednotka).

Technologické uzly linky, které jsou popsány v kapitole 4, mají vlastní POU, která obsahuje program zajišťující požadovanou funkčnost uzlu na základě jeho vlastností. Technologické uzly vykonávající rotační pohyb, například koncové navíjení a rozvádění, obsahují navíc podřízenou POU. Podřízená POU obsahuje ovládací datovou strukturu pro vykonání rotačního pohybu osy. Přehled technologických uzlů a jejich POU, najdeme v následující tabulce (Tabulka 5).

Na základě zjištěných poznatků z těchto POU, jakou jsou například: ovládací proměnné, ovládací parametry, vstupní parametry a výstupní parametry, bylo možné navrhnout požadovaný algoritmus pro řízení stroje v automatickém režimu. V kapitole 5.1 si nastíníme právě tyto základní ovládací parametry, které jsou nutné pro návrh algoritmu a nezbytné pro následné vytvoření softwaru pro řízení stroje v automatickém režimu.

Tabulka 5: Tabulka technologický uzlů a odpovídajících programových objektů

<b>Technologický uzel</b>	<b>Název POU</b>	<b>Název podřízených POU</b>
VN zdroj	<i>VN_zdroj</i>	-
Zákrut 1	<i>zakrut</i>	-
Zákrut 2	<i>Zakrut2</i>	-
Koncové navíjení a rozvádění	<i>Navijeni_Rozvadeni</i>	<i>Axis01Control, Axis02Control</i>
Ohřev 1	<i>ohrev_1</i>	-
Ohřev 2	<i>ohrev_2</i>	-
Dmýchadlo	<i>dmychadlo</i>	-
Peristaltické pumpy	<i>pumpy</i>	-
Šnekové pumpy	<i>Pumpy_snekove</i>	<i>Axis05Control, Axis06Control</i>
Navíjení v komoře	<i>navijeni_komora</i>	<i>Axis03Control</i>
Rozvádění v komoře	<i>Rozvadeni_komora</i>	<i>Axis04Control</i>



## 5.1 Parametry pro koncové navíjení a rozvádění

Pro demonstraci byly zvoleny parametry pro technologický uzel navíjení a rozvádění příze, který zakončuje technologický proces výroby nanovláken na pohybující se jádrovou přízi. Ovládací parametry jsou parametry, které musí uživatel nastavit, aby bylo možné spustit navíjení a rozvádění příze. Tyto parametry mají v *Automation Studiu* jiný název a najdeme je v globálních nebo lokálních proměnných.

- **Úhel křížení** - `gParam.Navin.uhel_krizeni_deg`

Protože se příze neklade na cívku rovnoběžně, nýbrž pod určitým úhlem, je zapotřebí znát úhel křížení, tak aby každý nový návin překřížil předcházející vrstvu příze. V *Automation Studiu* odpovídá úhlu křížení parametr, který se nachází mezi globálními proměnnými s označením `gParam.Navin.uhel_krizeni_deg`.

- **Soukací rychlost** - `SoukaciRychlost_setpoint`

Soukací rychlost je rychlost, se kterou se příze navíjí na cívku.

Důležité řídicí parametry, pomocí kterých spustíme navíjecí ústrojí:

- **Spražené ovládání** - `gParam.rozvadeni.ovl_navijenim`

Nejdříve je potřeba zapnout sprážené ovládání, aby po zapnutí rozvádění se zapnulo i navíjení.

- **Zapnutí rozvádění** - `gRozvadeni.cmd.run`

Zapnutím rozvádění spustíme proces navíjení příze na cívku.

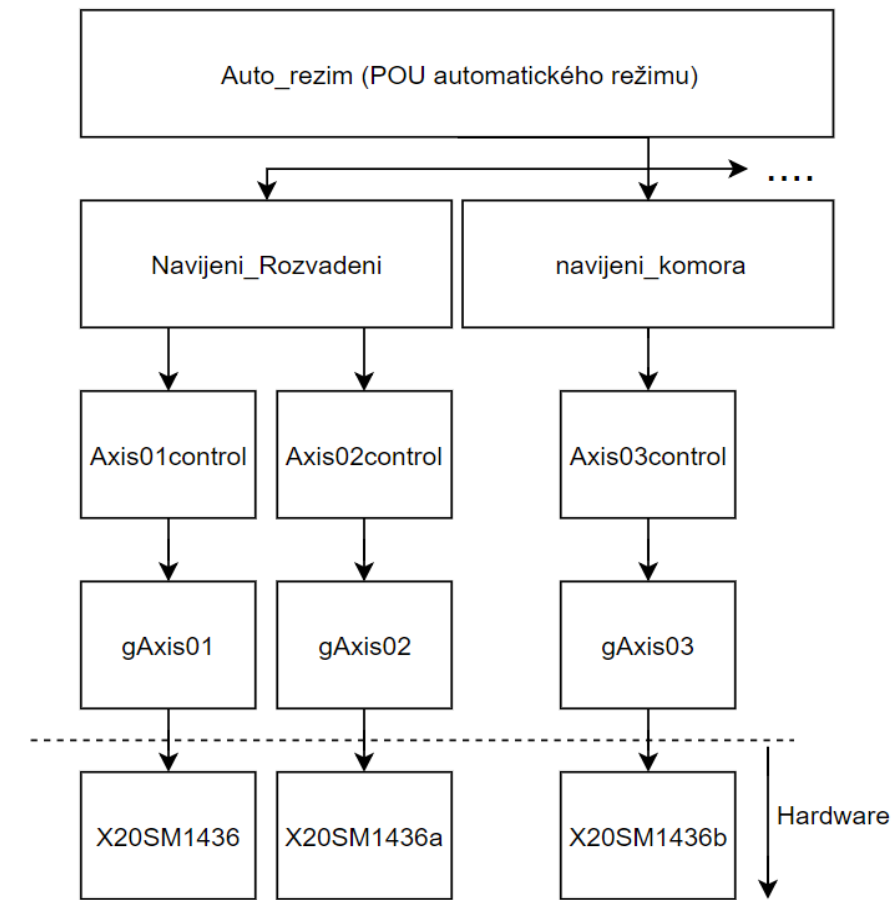
Další důležité parametry a proměnné:

- `gAxis01ctrl` - ovládací datová struktura pro osu navíjení,
- `gAxis02ctrl` - ovládací datová struktura pro osu rozvádění,
- `gAxis02ctrl.Status.ActPosition` - aktuální poloha osy rozvádění [mm],
- `gParam.rozvadeni.offset` – parametr pro offset rozvádění,
- `gAxis01ctrl.Command.Stop` – povel k zastavení osy navíjení,
- `gAxis02ctrl.Command.Stop` – povel zastavení osy rozvádění,
- `gAxis01ctrl.Command.Power` – povel ke spuštění navíjecí osy,
- `gAxis02ctrl.Command.Power` – povel ke spuštění rozváděcí osy.

## 5.2 Řídící struktura experimentální linky

Na následujícím obrázku (Obrázek 23) je vyobrazena hierarchie řízení experimentální linky. POU *Auto\_rezim* ovládá řídicí parametry v *Navijeni\_Rozvadeni*, *navijeni\_komora*, *zakrut*, *rozvadeni\_komora*, *pumpy*, *dmychadlo*, *ohrev\_1*, atd.

Programový blok vytvořený v POU *Navijeni\_Rozvadeni*, pracuje se základními řídicími parametry, které jsou uloženy v *gAxis01* a *gAxis02*. Tyto parametry pak ovládají přes řídicí modul *X20SM143* krokové motory navíjecího a rozváděcího ústrojí. Na stejném principu pracují i další POU.



Obrázek 23: Diagram řízení technologických uzlů

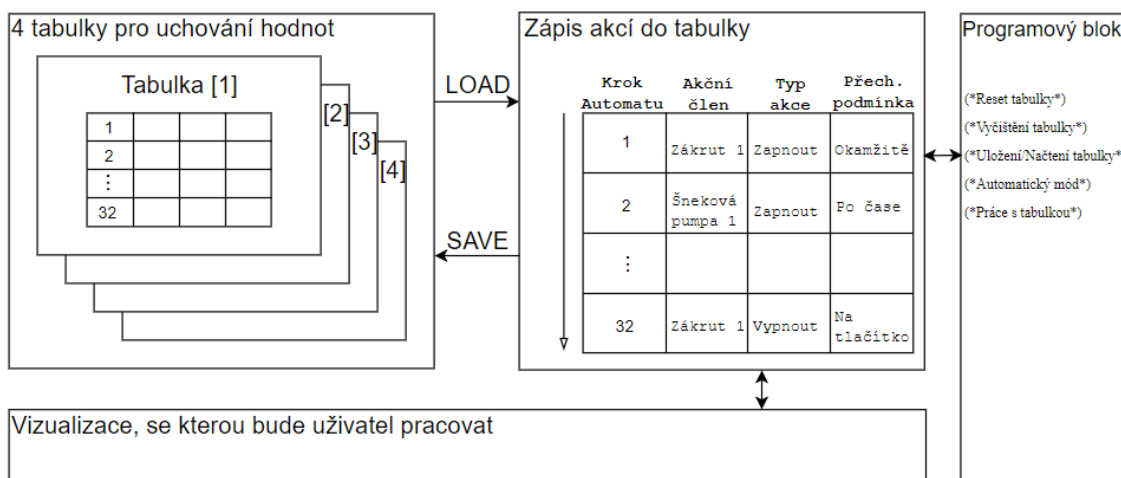
## 5.3 Vývojový diagram a struktura řízení

Před samotnou realizací vývojového diagramu a programového bloku, bylo potřeba promyslet a koncipovat strukturu řízení automatického režimu.

Kvůli možnosti experimentálního provozu, zapínání a vypínání jednotlivých uzlů v závislosti na situaci stroje, bylo zapotřebí použít tabulku. Tabulka slouží k definování posloupnosti zapínání respektive vypínání technologických uzlů dle požadavků konkrétní výrobní technologie.

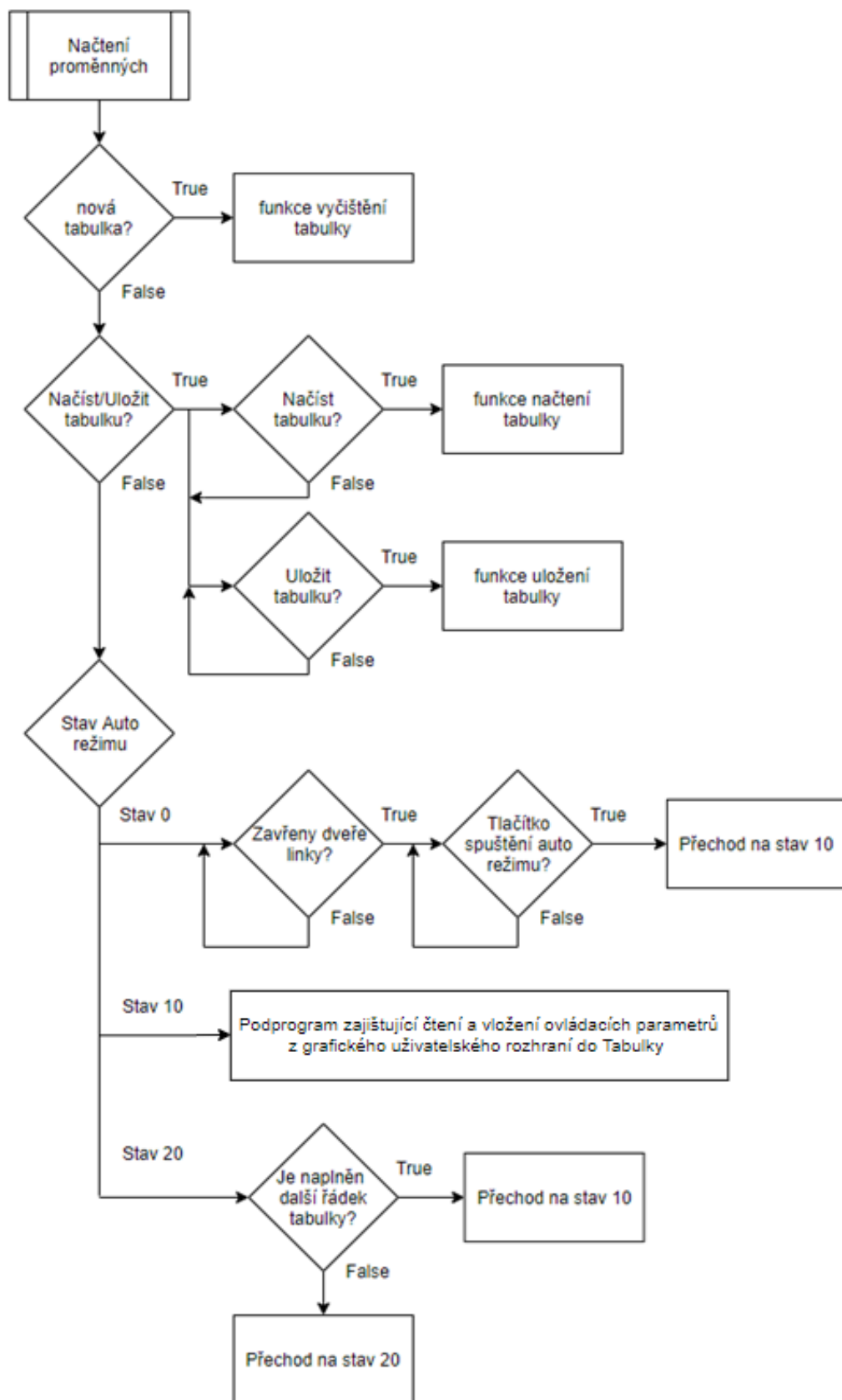
Pro možnost poloprovozního chodu, zamezení nežádoucích chyb od obsluhy, bylo potřeba předdefinovaného výrobního procesu, kde není zapotřebí příliš mnoho zásahů od obsluhy. Jinými slovy možnost nahrát předem uložený a ověřený výrobní proces.

V poslední řadě, kvůli snadnější práci s tabulkou, bylo potřeba zahrnout možnost resetování výrobního procesu a vyčištění tabulky pro případnou změnu procesu. Pro snadnější orientaci čtenáře v problematice automatického režimu, byl vytvořen diagram zobrazený na obrázku (Obrázek 24), znázorňující strukturu řízení.



Obrázek 24: struktura řízení automatického režimu

Ze znalosti struktury řízení bylo možné vytvořit vývojový diagram algoritmu, zajišťující požadovaný způsob použití automatického režimu stroje. Vývojový diagram algoritmu (viz Obrázek 25) zobrazuje, jakým způsobem je programový blok strukturován.



Obrázek 25: Vývojový diagram

## 5.4 Programový blok automatického režimu

Na základě vývojového diagramu (viz Obrázek 25) byl vytvořen programový blok automatického režimu. Pro vytvoření zdrojového kódu byl použit jako programovací jazyk strukturovaný text (viz 2.6.2). Program je tvořen především pomocí příkazu `CASE`, který nám zajišťuje větvení do různých stavů a pomocí podmínek `IF`. Protože PLC pracuje v programové smyčce, nemusí se programátor PLC starat o to, aby po konci programu vrátil jeho vykonávání opět na začátek. To zajistí již systémový program.

### 5.4.1 Použití globální proměnné, struktury a datové typy

Po spuštění programu jsou nejprve načteny globální a lokální proměnné a vlastní datové typy a struktury.

Pomocí datových typů a struktur vytvoříme tabulku, do které budeme ukládat uživatelem navolený výrobní proces. Přesněji řečeno technologické uzly tak, jak mají jít podle zvoleného výrobního procesu za sebou a jak se mají chovat a spouštět.

```
TYPE
  T_ActionTable :ARRAY[1..32]OF T_actionTableRow; (*tabulka s akcnimi clenymi*)

  T_actionTableRow :   STRUCT
    AC_number :   UINT; (*typ akcniho clenu*)
    Action_type :   UINT; (*typ akce s akcnim clenem*)
    Wait_time :   REAL; (*[m] - cas casovace Minuty*)
    Wait_times :   REAL; (*[s] - cas casovace Vteřiny*)
    Par01 :   REAL; (*parametr 1 pro AC*)
    StepCond :   INT; (*podminka přechodu na další AC*)
    NextRow :   UINT; (*další AC*)
    AC_Color :   UINT; (*Zvýraznění aktualního akčního členu*)
    AC_errorS :   BOOL; (*Chyba akčního členu*)
  END_STRUCT;
  T_status :   STRUCT
    ready :   BOOL; (*auto režim možno spustit*)
    error :   BOOL; (*auto režim v chybě*)
    stopping :   BOOL; (*auto režim zastavuje*)
    run :   BOOL; (*auto režim v chodu*)
    starting :   BOOL; (*auto režim se rozbíhá*)
    stop :   BOOL; (*auto režim stojí*)
  END_STRUCT;
END_TYPE
```

*Zdrojový kód 1: Datové typy a struktury*

Popis datových typů a struktur zobrazených ve zdrojovém kódu (Zdrojový kód 1):

- **T\_ActionTable**

Datový typ `T_ActionTable` je typu `ARRAY` o rozměru 32 řádků (odpovídá počtu kroků v tabulce) a vnořenou strukturou `T_actionTableRow` (odpovídá sloupcům v tabulce).

- **T\_actionTableRow**

Datový typ `T_actionTableRow` je typu `STRUCT` a umožňuje spojení několika různých datových typů do jednoho a pracovat s nimi jako s celkem. V našem případě například:

- `AC_number` - je typu `UINT` a obsahuje, s jakým akčním členem neboli technologickým uzlem chceme pracovat.
- `Action_type` - je typu `UINT` a nastavuje, jaký typ akce se s daným akčním členem vykoná.
- `Wait_time` - je typu `REAL` a nastavuje časovač na základě zvolené přechodové podmínky.
- `Par01` - je typu `REAL` a nastavuje parametr aktuálního akčního členu, například otáčky motoru.
- `StepCond` - je typu `INT` a nastavuje přechodovou podmínku pro skok na další řádek v tabulce.

- **T\_status**

Datový typ `T_status` je typu `STRUCT` a obsahuje, v jakém režimu se automatický chod právě nachází.

Popis globálních proměnných zobrazených v následujícím zdrojovém kódu (Zdrojový kód 2):

- **AC\_TableR**

`AC_TableR` je typu `ARRAY` a jedná se o dvourozměrné pole o rozměru 4x32. Proměnná je typu `RETAIN`, která zůstává v paměti i po vypnutí stroje. Slouží k uložení tabulky pro případné pozdější opakované použití.

- **Konstanty pro AC\_number**

Jedná se o konstanty, které odpovídají akčním členům neboli technologickým uzlům stroje. Z těchto konstant si uživatel volí v každém kroku programu, podle toho, se kterým členem chce zrovna pracovat. Výběr se uloží do proměnné `AC_number`.

- **Konstanty pro Action\_type**

Slouží pro zvolení akce s daným akčním členem. Například zapnutí nebo vypnutí pump.

- **Konstanty pro StepCond**

Konstanty znázorňující s jakou přechodovou podmínkou skočíme na další řádek v tabulce.

```

VAR
  AC_Table : ARRAY[1..32] OF T_actionTableRow; (*Tabulka pro automatický
režim*)
  AC_ActRow : T_actionTableRow;
  Auto_status : T_status := (0); (*stav automatického režimu*)
  AC_Index : UINT := 1; (*Radek tabulky*)
  AC_IndexR : UINT := 0; (*Radek tabulky*)
  AC_state : USINT := 0; (*stav automatickeho rezimu*)
  TransTimer : TON; (*Casovac*)
  AC_Reset : BOOL := FALSE; (*Reset tabulky na začátek automatického
režimu*)
  AC_Clean : BOOL := FALSE; (*Vymazání formuláře automatického režimu*)
  AC_Load : INT := 0; (*Uložení formuláře AC do předvoleb*)
  AC_Save : INT := 0; (*Uložení formuláře AC do předvoleb*)
  AC_SaveLoad : INT := 0; (*Uložení formuláře AC do předvoleb*)
  AC_SavePopUp : INT := 1; (*PopUp okno pro AC předvolby*)
  AC_Zapnout : BOOL := FALSE; (*Zapnutí Vypnutí automatickeho rezimu*)
  Second : TIME;
  Minute : TIME;
  Time1 : TIME;
END_VAR
(*Možnost uložení tabulky *)
VAR RETAIN
  AC_TableR : ARRAY[1..4,1..32] OF T_actionTableRow; (*uložení/načtení
tabulky*)
END_VAR
(*KONSTANTY PRO AC_number*)
VAR CONSTANT
  AC_Nevybrano : UINT := 0;
  AC_Konec : UINT := 1;
  AC_Navijeni_komora : UINT := 2;
  AC_Rozvadeni_komora : UINT := 3;
  AC_Navijeni_Rozvadeni : UINT := 4;
  AC_Pumpy_snekove_1 : UINT := 5;
  AC_Pumpy_snekove_2 : UINT := 6;
  AC_Pumpy_preristalticke_1 : UINT := 7;
  AC_Pumpy_preristalticke_2 : UINT := 8;
  AC_Zakrut_1 : UINT := 9;
  AC_Zakrut_2 : UINT := 10;
  AC_Ohrev_1 : UINT := 11;
  AC_Ohrev_2 : UINT := 12;
  AC_VN_zdroj : UINT := 13;
  AC_Dmychadlo : UINT := 14;
END_VAR
(*KONSTATY PRO Action_type*)
VAR CONSTANT
  ACT_Nevybrano : UINT := 0; (*Nevybrano*)
  ACT_Zapnout : UINT := 1; (*Zapnutí*)
  ACT_Vypnout : UINT := 2; (*Vypnutí*)
END_VAR
(*KONSTANTY PRO STEP CONDITION*)
VAR CONSTANT
  ACSC_Nevybrano : UINT := 0; (*Nevybrano*)
  ACSC_Immediately : UINT := 1; (*Okamžitý přechod*)
  ACSC_Wait_time : UINT := 2; (*Přechod po určitém čase*)
  ACSC_Value_reached : UINT := 3; (*Přechod po dosažení požadované
hodnoty*)
  ACSC_Button : UINT := 4; (*Přechod na tlačítko Start*)
  ACSC_ButtonOff : UINT := 5; (*Přechod na tlačítko Stop*)
END_VAR

```

*Zdrojový kód 2: Globální proměnné*

## 5.4.2 Vyčištění tabulky

V každém cyklu programu dochází k dotazu, zda nedošlo ke stisknutí tlačítka pro vyčištění tabulky pomocí funkce `IF`. V případě že ano, dojde nejprve k zastavení všech běžících uzlů a poté k defaultnímu nastavení všech prvků v tabulce. Pomocí funkce `FOR` nastavíme pro každý řádek tabulky s náležitou vnořenou proměnnou hodnotu „0“ která odpovídá defaultní volbě. Funkce zajišťující vyčištění tabulky najdeme v následujícím zdrojovém kódu (Zdrojový kód 3).

```
(*Vyčištění tabulky*)
IF AC_Clean THEN
  gZakrut.command.Power := FALSE;
  gZakrut2.command.Power := FALSE;
  gNavijeni.cmd.run := FALSE;
  gRozvadeni.cmd.run := FALSE;
  gVN_zdroj.cmd.run := FALSE;
  gOhrev_1.cmd.enable := FALSE;
  gOhrev_2.cmd.enable := FALSE;
  gDmychadlo.cmd.enable := FALSE;
  gPumpy.cmd.run_1 := FALSE;
  gPumpy.cmd.run_2 := FALSE;
  gPumpSnek_1.cmd.run := FALSE;
  gPumpSnek_2.cmd.run := FALSE;
  gRozvadeni_komora.cmd.run := FALSE;
  gNavijeni_komora.cmd.run := FALSE;
  Auto_status.stop := TRUE;
  Auto_status.ready := FALSE;
  Auto_status.stopping := FALSE;
  Auto_status.starting := FALSE;
  Auto_status.run := FALSE;

  FOR AC_Index := 0 TO 32 BY 1 DO
    AC_Table[AC_Index].AC_Color := 0;
    AC_Table[AC_Index].Action_type := 0;
    AC_Table[AC_Index].AC_number := 0;
    AC_Table[AC_Index].Par01 := 0;
    AC_Table[AC_Index].StepCond := 0;
    AC_Table[AC_Index].Wait_time := 0;
    AC_Table[AC_Index].Wait_timeS := 0;
    AC_Table[AC_Index].AC_errorS := FALSE;
  END_FOR;
  AC_SavePopUp := 1;
  AC_Index := 1;
  AC_state := 0;
  AC_Clean := FALSE;
  AC_Zapnout := FALSE;
END IF;
```

*Zdrojový kód 3: Funkce pro vyčištění tabulky*

## 5.4.3 Uložení a načtení tabulky

Stejně jako u vyčištění tabulky, tak i zde je potřeba v každém cyklu kontrolovat, zda nedošlo ke stisknutí tlačítka pro uložení a načtení tabulky. Stisknutí tlačítka vyvolá vyskakovací okno, kde najdeme 4 předvolby a tlačítka pro uložení nebo načtení. Stisknutím tlačítka pro uložení, dojde ke zkopírování aktuální tabulky do tabulky `AC_TableR`, která je typu `RETAIN` a uchová její hodnotu i po vypnutí stroje.



Stisknutím tlačítka načíst dojde k obrácenému efektu a to ke zkopírování tabulky *AC\_TableR* do aktuální tabulky.

Funkce, zajišťující načtení a uložení tabulky (Zdrojový kód 4).

```
(*Uložení a Načtení tabulky*)
IF AC_SavePopUp = 0 THEN
  IF AC_Save = 1 THEN
    FOR AC_IndexR := 0 TO 32 BY 1 DO
      AC_TableR[AC_SaveLoad,AC_IndexR].Action_type := AC_Table[AC_IndexR].Action_type;
      AC_TableR[AC_SaveLoad,AC_IndexR].AC_number := AC_Table[AC_IndexR].AC_number;
      AC_TableR[AC_SaveLoad,AC_IndexR].Par01 := AC_Table[AC_IndexR].Par01;
      AC_TableR[AC_SaveLoad,AC_IndexR].StepCond := AC_Table[AC_IndexR].StepCond;
      AC_TableR[AC_SaveLoad,AC_IndexR].Wait_time := AC_Table[AC_IndexR].Wait_time;
      AC_TableR[AC_SaveLoad,AC_IndexR].Wait_timeS := AC_Table[AC_IndexR].Wait_timeS;
    END_FOR;
    AC_Save := 0;
    AC_SaveLoad := 0;
    AC_SavePopUp := 1;
  END_IF;
  IF AC_Load = 1 THEN
    FOR AC_IndexR := 0 TO 32 BY 1 DO
      AC_Table[AC_IndexR].Action_type := AC_TableR[AC_SaveLoad,AC_IndexR].Action_type;
      AC_Table[AC_IndexR].AC_number := AC_TableR[AC_SaveLoad,AC_IndexR].AC_number;
      AC_Table[AC_IndexR].Par01 := AC_TableR[AC_SaveLoad,AC_IndexR].Par01;
      AC_Table[AC_IndexR].StepCond := AC_TableR[AC_SaveLoad,AC_IndexR].StepCond;
      AC_Table[AC_IndexR].Wait_time := AC_TableR[AC_SaveLoad,AC_IndexR].Wait_time;
      AC_Table[AC_IndexR].Wait_timeS := AC_TableR[AC_SaveLoad,AC_IndexR].Wait_timeS;
    END_FOR;
    AC_Load := 0;
    AC_SaveLoad := 0;
    AC_SavePopUp := 1;
  END_IF;
  AC_Zapnout := FALSE;
END_IF;
```

*Zdrojový kód 4: Funkce pro uložení a načtení tabulky*

#### 5.4.4 Přejít mezi stavy automatického režimu

Automatický režim je rozdělen do tří stavů, které odpovídají fázi, v jaké se stroj nachází. Větvení do jednotlivých stavů zajišťuje příkaz **CASE**. Jednotlivé stavy jsou zobrazeny ve vývojovém diagramu (viz Obrázek 25).

Popis jednotlivých stavů:

- **Stav 0**

Ve stavu 0 probíhá zjištění, zda jsou spuštěné ovládací obvody a uzavřená dvířka zvláknovací komory. Pokud nejsou, zobrazí se na uživatelském rozhraní chyba, upozorňující na otevřená dvířka nebo vypnuté ovládací obvody. V opačném případě se režim stroje změní na „READY“ a vyčkává se na spuštění automatického režimu. Ke spuštění může dojít za pomoci dvou tlačítek, jedno je hardwarové a druhé softwarové. V případě sepnutí tlačítka se režim stroje změní na „STARTING“ a dojde ke změně stavu na stav 10.

- **Stav 10**

Nejprve dojde k přiřazení barvy k aktuálnímu a předchozímu řádku v tabulce, pro lepší orientaci uživatele v uživatelském rozhraní stroje. Poté se uživatelem nastavený čas přepočítá na milisekundy, aby bylo možné tento čas zadat do *Timeru*. *Timer* má zde funkci časovače, pro případ přechodu na další řádek tabulky po určitém čase.

Následuje nejdůležitější část programu a to čtení uživatelem navoleného výrobního procesu a následné zapsání jednotlivých kroků do řádků tabulky. Zápis do tabulky si popíšeme v kapitole 5.5.

- **Stav 20**

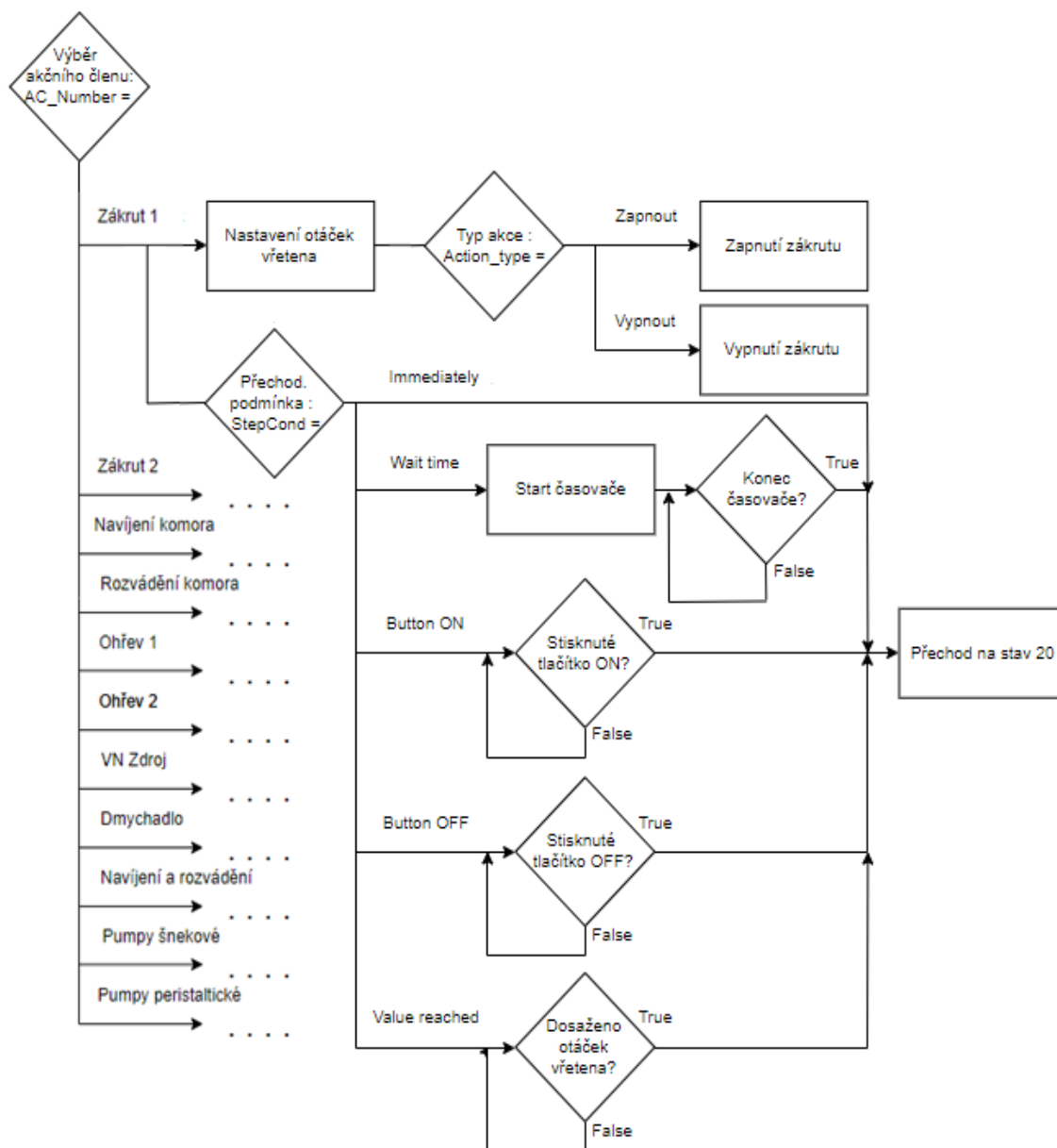
Ve stavu 20 dochází k rozpoznání, v jakém režimu se automat nachází. Pokud je následující řádek prázdný a všechny uzly jsou vypnuté, nastaví se režim stroje na „STOP“. Pokud je však alespoň jeden uzol zapnutý, změní se režim stroje na „RUN“. Dále dochází k rozpoznání, zda se automatický režim spouští „STARTING“, nebo zastavuje „STOPPING“.

## 5.5 Čtení a zápis dat tabulky

Po spuštění automatického režimu dojde ke čtení navolené produkce z grafického uživatelského rozhraní a následnému zapsání do tabulky. Program projde tabulku a podle zvoleného akčního členu vykoná požadované akce. Ovládací parametr, ať už se jedná o teplotu dmychadla nebo například rychlost vřeten, se zapíše z tabulky do odpovídající proměnné daného technologického uzlu.

Podle zvolené akce dojde k zapnutí nebo zastavení akčního členu a přechodová podmínka zajistí požadovaný přechod na další řádek v tabulce. Okamžitý přechod skočí ihned na další řádek tabulky. Přechod po určitém čase nastaví časovač podle uživatelem navoleného času a skočí až po uplynutí času. Podmínky přechodu na tlačítko čekají, dokud nedojde ke stisknutí hardwarového nebo softwarového tlačítka. Poslední možností je přechod po dosažení požadované hodnoty, tuto možnost neposkytují všechny akční členy. Například u zákrotu se čeká na dosažení požadovaných otáček nebo u ohřevu na dosažení požadované teploty.

Na následujícím obrázku (Obrázek 26) je znázorněn vývojový diagram pro akční člen „Zákrut 1“ a v následujícím zdrojovém kódu (Zdrojový kód 5) je vyobrazena část programu pro zákrut, který odpovídá tomuto vývojovému diagramu.



Obrázek 26: Vývojový diagram pro práci s tabulkou

```

AC_Zakrut_1 : (*pracujeme se zakrutem 1*)

(*Nastavení parametrů*)
gParam.zakrut.otacky_vreteno := AC_Table[AC_Index].Par01; (*Otáčky*)
AC_Table[AC_Index].AC_errorS := gZakrut.status.Error;

CASE AC_Table[AC_Index].Action_type OF (*požadavek na zapnutí ci vypnutí AC*)
    ACT_Zapnout : (*požadavek na zapnutí zakrutu 1*)
        gZakrut.command.Power := TRUE;
    ACT_Vypnout : (*požadavek na vypnutí zakrutu 1*)
        gZakrut.command.Power := FALSE;
END_CASE;

CASE AC_Table[AC_Index].StepCond OF (*vyhodnocení podmínky přechodu na další řádek tabulky*)
    ACSC_Immediately : (*okamžitý přechod na další AC*)
        AC_Index := AC_Index + 1;
        AC_state := 20;
    ACSC_Wait_time :
        TransTimer.PT := Time1;
        TransTimer.IN := TRUE;

        IF TransTimer.Q THEN
            AC_Index := AC_Index + 1;
            AC_state := 20;
            TransTimer.IN := FALSE;
        END_IF;
    ACSC_Button :
        IF AC_Zapnout OR di.start THEN
            AC_Index := AC_Index + 1;
            AC_state := 20;
            AC_Zapnout := FALSE;
        END_IF;
    ACSC_Value_reached :
        IF gZakrut.status.TargetReached THEN
            AC_Index := AC_Index + 1;
            AC_state := 20;
        END_IF;
    ACSC_ButtonOff :
        IF AC_Zapnout OR NOT di.stop THEN
            AC_Index := AC_Index + 1;
            AC_state := 20;
            AC_Zapnout := FALSE;
        END_IF;
END_CASE;

```

*Zdrojový kód 5: Část programu pro Zákrut 1*

## 6 Grafické uživatelské rozhraní

Pro tvorbu uživatelského grafického rozhraní slouží software *VC Visualisation*, který je součástí produktu *Automation Studio*. Pracovní prostředí pro tvorbu vizualizace je zobrazeno v kapitole 2.4.

Vizualizační prostředí bylo vytvořeno s rozlišením 1024x768 (XGA) navzdory grafickému rezistivnímu 5.7" dotykovému panelu o maximálním rozlišení 320x240 (QVGA). Učiněno bylo z nedostatku místa pro vytvoření tabulky automatického režimu a z velmi náročného ovládání stroje v malém rozlišení panelu. Ovládání uživatelského rozhraní experimentální linky je v tuto chvíli možné pouze přes VNC (Virtual Network Computing) aplikace třetích stran z osobního počítače. Použití VNC aplikace je zapotřebí kvůli běžícímu VNC serveru (server zajišťující grafické zobrazení) na PLC. Řešením do budoucna, pro možnost ovládání stroje z ovládacího panelu, je pořízení nového dotykového panelu podporující rozlišení XGA.

Při tvorbě nové vizualizace bylo potřeba zahrnout i možnost manuálního režimu ovládání. Vizualizace tedy obsahuje záložky se všemi technologickými uzly, jejich ovládacími parametry a možností spustit každý uzel samostatně.

Uživatelské rozhraní bylo rozděleno do dvou navigačních panelů po stranách kvůli přehlednosti. Na pravé straně se nachází hlavní navigační panel, který obsahuje 6 záložek a to: *Hlavní strana*, *Trendy*, *Alarmy*, *Diagnostika*, *Auto režim* a *Nastavení*. Nejdůležitější z nich si popíšeme zde:

- ***Hlavní strana***

Na hlavní straně vizualizace (Obrázek 27) jsou zobrazeny, z důvodu přehlednosti, všechny výstupní parametry pohromadě. Jedná se o teploty ohřevů, teploty ve zvláknovací komoře, otáčky zákrutů, otáčky navíjení a rozvádění, atd.

- ***Alarmy***

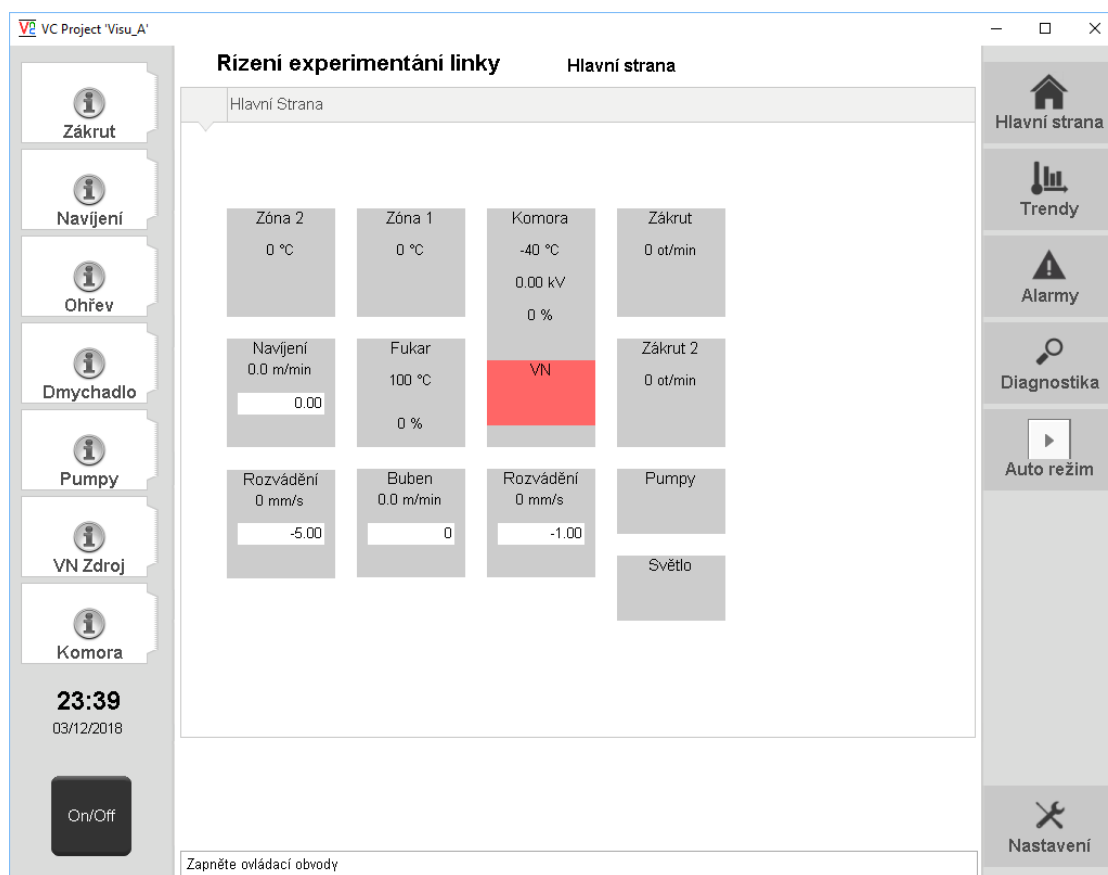
Na stránce s alarmy najdeme výpis chyb, ke kterým došlo.

- ***Auto režim***

Stránka pro nastavení automatického režimu. Popíšeme si v samostatné kapitole 6.1.

- **Nastavení**

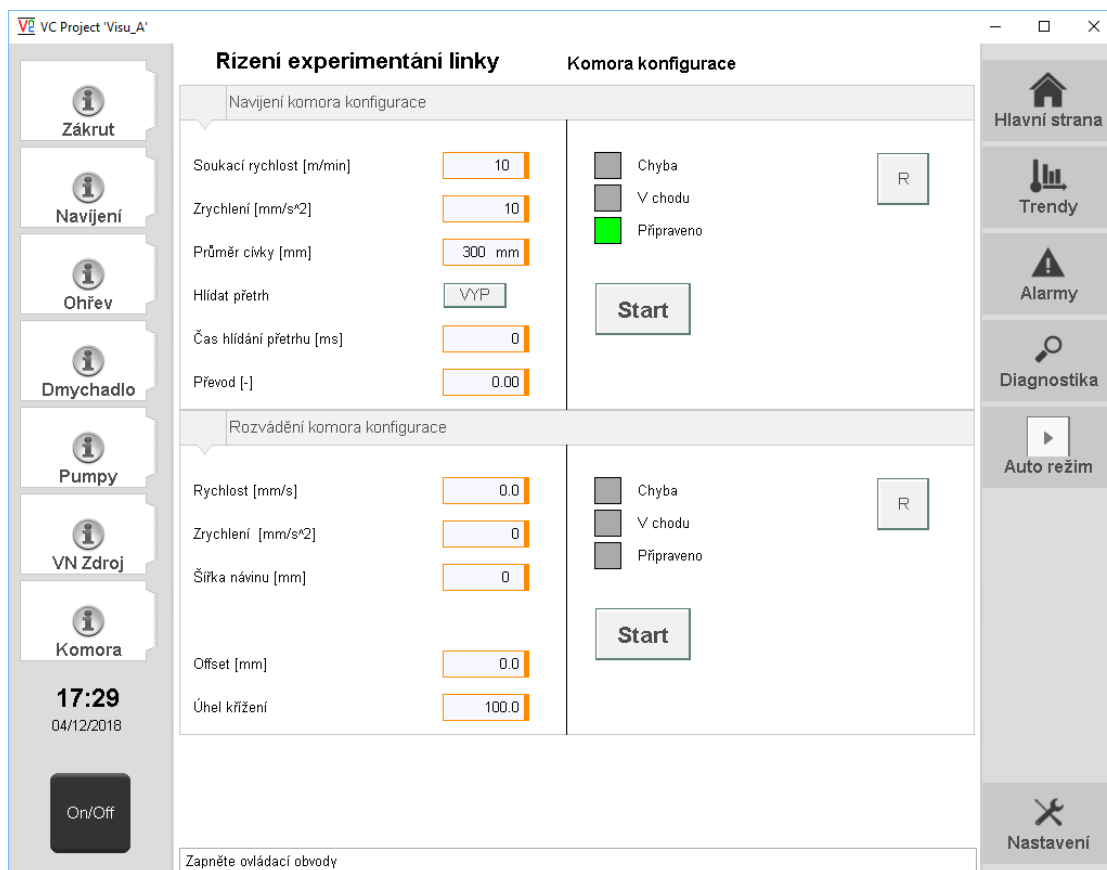
Nejdůležitějším nastavením je možnost změny jazyka na angličtinu, pro možnost práce na zařízení zahraničních studentů. Dále je zde možnost konfigurace dotyku, jasu, kontrastu a možnost nastavení data a času.



Obrázek 27: Hlavní strana uživatelského rozhraní

Na levé straně se nachází navigační panel se záložkami technologických uzlů pro možnost manuálního režimu. Jedná se o záložky: *Zákruť*, *Navíjení*, *Ohřev*, *Dmychadlo*, *Pumpy*, *VN Zdroj* a *Komora*. Technologické uzly jsou popsány v kapitole 4. Na následujícím obrázku (Obrázek 28) je zobrazena záložka *Komora* a obsahuje ovládání pro navíjení a rozvádění v komoře.

Technologické uzly, které spolu souvisí, jsou sloučeny do jedné záložky pro lepší přehlednost. Například záložka *Navíjení*, obsahuje ovládání pro koncové navíjení a rozvádění. Kromě ovládacích prvků obsahují záložky i možnost nastavení veškerých parametrů, ať už se jedná o rychlost, zrychlení, šířku návinnu, úhel křížení, průměr cívky, průtok vzduchu, atd. V poslední řadě najdeme i signalizaci stavu a tlačítko pro resetování, když se technologický uzel nachází v chybě.



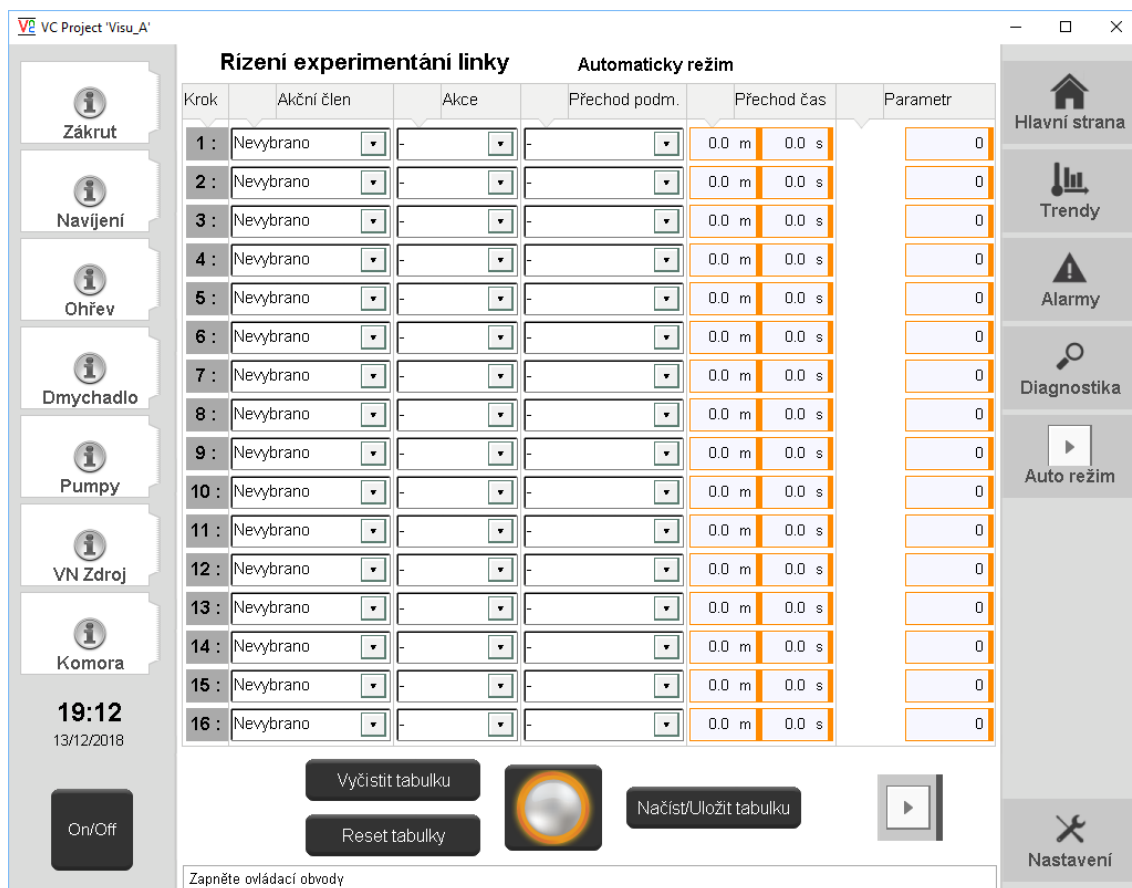
Obrázek 28: Navijení a rozvádění v komoře

## 6.1 Auto režim

Stránka pro nastavení automatického režimu byla realizována tak, aby obsluha mohla snadno zadat postup výroby dle aktuálního mechanického uspořádání linky.

Postup výroby se zadává do tabulky, která má 32 řádků, tak jak požaduje obsluha, aby se mechanické části linky spouštěli za sebou. Do sloupců ke každému řádku uživatel nastaví, s jakým uzlem si přeje pracovat, co se má vykonat a s jakou podmínkou se má přejít na další řádek. V Případě že uživatel zvolí přechod po určitém čase, musí nastavit v dalším sloupci určený čas. Pro nastavení času jsou určeny dvě kolonky, kde do první se zadávají minuty a do druhého sekundy. V poslední řadě je potřeba nastavit hlavní parametr, ten se liší podle zvoleného uzlu. Například v případě že se jedná o navijení, je hlavním parametrem soukací rychlost. Pokud se jedná o rozvádění je parametrem rychlost rozvádění.

Hlavní parametry a jakému prvku odpovídají, si ukážeme v kapitole 7. Stránka s automatickým režimem je na následujícím obrázku (Obrázek 29).



Obrázek 29: Automatický režim

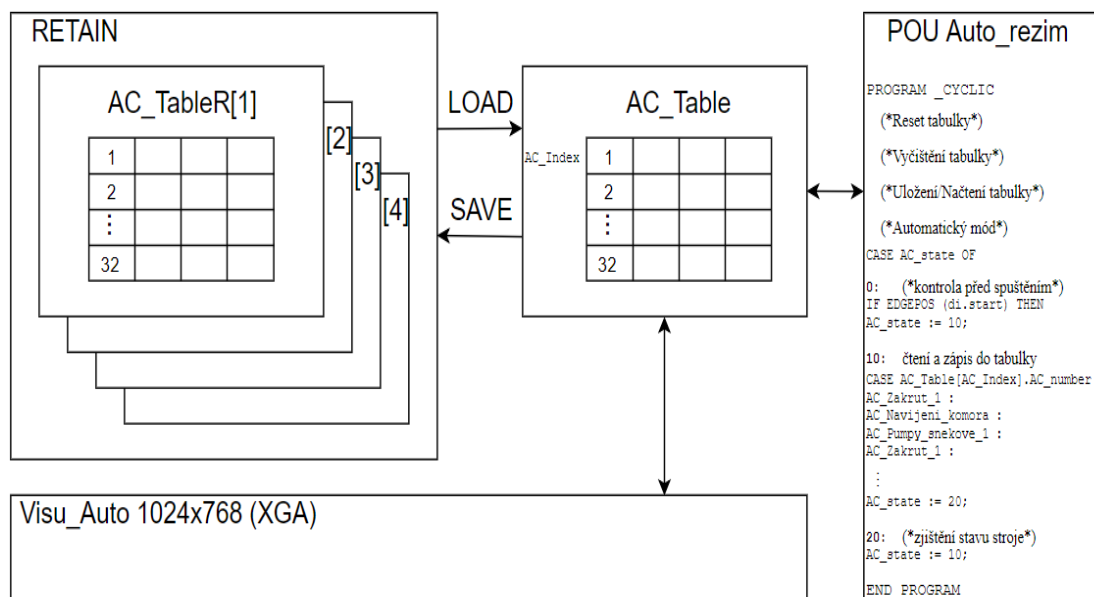
Kromě tabulky kam se zadává postup výroby, stránka obsahuje ještě softwarové tlačítko, které se nachází uprostřed dole. Tlačítko slouží jako pomocné, při podmínce přechodu na tlačítko. Například v případě že obsluha chce přejít na další akční člen až v případě, kdy vizuálně zhodnotí situaci stroje a nechce používat hardwarová tlačítka na ovládacím panelu.

Dalším tlačítkem na stránce automatického režimu je „Vyčistit tabulku“. Toto tlačítko nejprve zastaví všechny běžící uzly, aby nemohly nastat chyby. Poté nastaví tabulku a automatický režim do výchozího nastavení. Výchozí nastavení odpovídá prvotnímu spuštění vizualizace (viz Obrázek 29).

Tlačítko „Reset tabulky“ nastaví automatický režim do výchozího nastavení, avšak ponechá naplněnou tabulku se všemi akcemi a podmínkami. Pro případ když uživatel potřebuje spustit celý výrobní proces znovu od začátku.



Po stisknutí tlačítka „Načíst/Uložit tabulku“ vyskočí okno, kde jsou na výběr čtyři předvolby. Po stisknutí jedné z předvoleb a následného stisknutí tlačítka „uložit“ se provede nahrání aktuální uživatelem vyplněné tabulky do dané předvolby pro pozdější použití. V opačném případě, po zvolení předvolby a stisknutí tlačítka „Nahrát“, se již uložená tabulka v předvolbě nahraje do tabulky aktuální. Diagram, jakým způsobem pracuje tabulka s programem a s vizualizací, najdeme na obrázku (Obrázek 30).

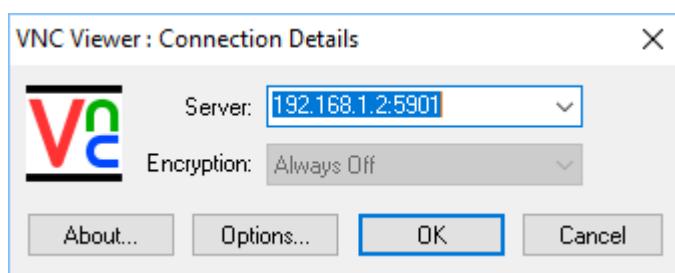


Obrázek 30: Diagram pro práci s tabulkou

## 7 Návod na použití automatického režimu

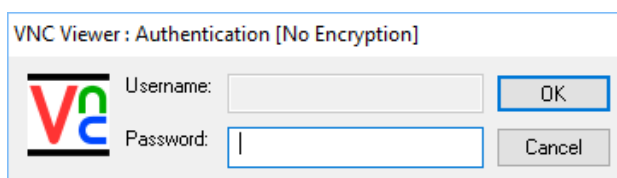
### 7.1. Připojení k vizualizaci

Vizualizační prostředí bylo vytvořeno s vyšším rozlišením (XGA), než grafický dotykový panelu podporuje. Učiněno bylo kvůli pohodlnějšímu ovládání a z nedostatku místa pro tabulku. Ovládání automatického režimu je v tuto chvíli možné jen přes VNC aplikace třetích stran z osobního počítače. Například přes aplikaci „VNC Viewer“, pomocí které se vzdáleně připojíme na PLC.



Obrázek 31: VNC Viewer

Do kolonky „Server“ napíšeme IP adresu vzdáleného VNC serveru běžícím na PLC. Za IP adresu **192.168.1.2** (Adresa VNC serveru) je nutné zadat port **5901**, na kterém budeme komunikovat s vizualizací pro automatický režim. Po stisknutí tlačítka „OK“, vyskočí okno kde je potřeba zadat heslo. Nativně jsou přiřazená písmena „w“ pro oprávnění zápisu a čtení a „r“ jen pro čtení.



Obrázek 32: VNC Viewer ověření

## 7.2. Popis automatického režimu

Tabulka automatického režimu obsahuje 32 řádků a uživatel ji může naplnit kombinací 14 akčních členů. Jednotlivé členy je potřeba vybrat z rozbalovací nabídky a přiřadit do prvního sloupce (*Akční člen*). Na následujícím obrázku (Obrázek 33) je zobrazena stránka s automatickým režimem a s vyznačeným tlačítkem pro rozbalení nabídky akčních členů. V následující tabulce (Tabulka 6) najdeme výpis všech akčních členů s informací, o jaký technologický uzel se jedná a v jaké záložce najdeme ostatní parametry s tímto uzlem spojené.

Obrázek 33: Automatický režim - rozbalovací nabídka s akčními členy

Tabulka 6: přehled akčních členů z rozbalovací nabídky

Akční člen	Záložka	Informace o technologickém uzlu
Navijeni komora	<i>Komora</i>	Navíjení plošného produktu na bubnový kolektor
Rozvadení komora	<i>Komora</i>	Rozvádění plošného produktu na bubnový kolektor
Navijeni Rozvadení	<i>Navijeni</i>	Koncové navíjení a rozvádění příže na cívku
Pump Snek 1	<i>Pumpy</i>	Šneková pumpa 1 pro dávkování polymerního roztoku
Pump Snek 2	<i>Pumpy</i>	Šneková pumpa 2 pro dávkování polymerního roztoku
Pump Perist 1	<i>Pumpy</i>	Peristaltická pumpa 1 pro dávkování polymeru

Pump Perist 2	<i>Pumpy</i>	Peristaltická pumpa 2 pro dávkování polymeru
Zakrut 1	<i>Zákrut</i>	Zákrutové zařízení před vstupem do zvláknovací komory
Zakrut 2	<i>Zákrut</i>	Zákrutové zařízení za výstupem ze sušících trubíc
Ohrev 1	<i>Ohřev</i>	Sušící trubice 1 sloužící k udržení požadované teploty
Ohrev 2	<i>Ohřev</i>	Sušící trubice 2 sloužící k udržení požadované teploty
VN zdroj	<i>VN zdroj</i>	Transformátor vysokého napětí pro zvláknovací proces
Dmychadlo	<i>Dmychadlo</i>	Horkovzdušné dmychadlo pro předehřátý vzduch do trubíc
Časovač	-	Funkce zajišťující prodloužení času výrobní produkce

Akční člen *Časovač* slouží pro prodloužení přechodového času na další řádek tabulky. Například když je zapotřebí, aby linka běžela několik hodin v kuse a po uplynutí této doby se započala vypínací sekvence, je možné použití více časovačů za sebou.

Po zvolení akčního členu, je zapotřebí vybrat jaká akce se s daným akčním členem má vykonat. Lze zvolit ze dvou možností:

- ***Zapnout***

Akce, která provede zapnutí vybraného akčního členu.

- ***Vypnout***

Akce, která provede vypnutí vybraného akčního členu.

Následující sloupec v tabulce obsahuje v rozbalovací nabídce přechodové podmínky. Tyto podmínky zajišťují jakým způsobem dojde k přechodu na další řádek v tabulce. Podmínek je na výběr hned několik, aby bylo možné dosáhnout požadovaného koncového produktu. Například když obsluha požaduje spuštění koncového navíjení a rozvádění jádrové příze až po dosažení požadované teploty v sušící trubici, použije jako přechodovou podmínku *Dosažení hodnoty*.

Podmínky přechodu:

- ***Okamžitý***

Okamžitý přechod na další řádek.

- ***Po čase***

Přechod po určitém čase, který uživatel navolí v dalším sloupci (Přechod čas). Sloupec obsahuje dvě pole pro vložení hodnoty, kde do prvního se zadávají minuty a do druhého vteřiny.

- **Dosažení hodnoty**

Přechod po dosažení požadované hodnoty (je možné nastavit jen u několika akčních členů). Jedná se o oba zákruty, kde se čeká na dosažení požadovaných otáček a oba ohřevy kde se čeká na dosažení požadovaných teplot.

- **Tlačítko Start**

Přechod po stisknutí hardwarového tlačítka „START“ které se nachází na ovládacím panelu.

- **Tlačítko Stop**

Přechod po stisknutí hardwarového tlačítka „STOP“ které se nachází na ovládacím panelu.

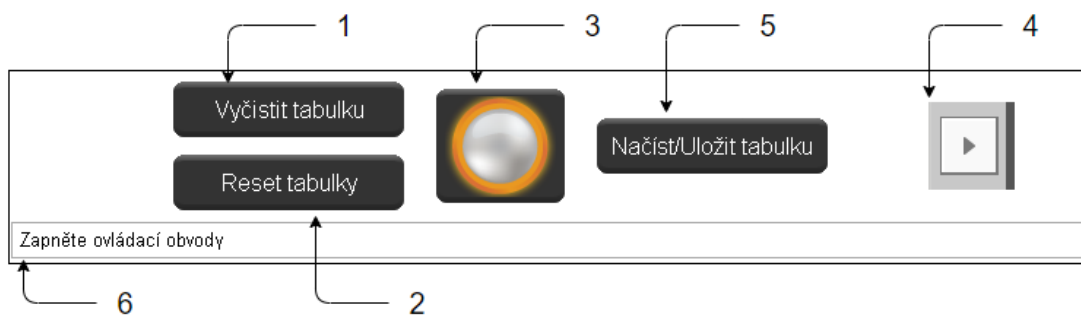
V poslední řadě, před samotným spuštěním automatického režimu, je zapotřebí nastavit hlavní parametr v posledním sloupci tabulky. Parametr odpovídá zvolenému akčnímu členu. Přehled akčních členů, u kterých je zapotřebí nastavit hlavní parametr a u kterých ne, najdeme v tabulce (Tabulka 7).

*Tabulka 7: Parametry akčních členů*

<b>Akční člen</b>	<b>Hlavní parametr</b>
Zákrut 1	Otáčky vřetena 1
Zákrut 2	Otáčky vřetena 2
Navíjení Rozvádění	Soukací rychlost příze
Ohřev 1	Teplota v trubici 1
Ohřev 2	Teplota v trubici 2
Pumpy Šnekové 1	Otáčky šnekové pumpy 1
Pumpy Šnekové 2	Otáčky šnekové pumpy 1
Pumpy Peristaltické 1	-
Pumpy Peristaltické 2	-
VN zdroj	-
Dmychadlo	Teplota vzduchu
Navíjení v komoře	Soukací rychlost
Rozvádění v komoře	Rychlost rozvádění

### 7.3. Navigační menu automatického režimu

Navigační lišta slouží pro manipulaci s tabulkou. Kromě tlačítek však obsahuje i informační panel, kterým nám ukazuje, v jakém stavu se stroj nachází. Informační panel je zobrazen vespod obrázku (Obrázek 34), kde nám zobrazuje chybovou zprávu, že nejsou zapnuté ovládací obvody. Kromě chybových zpráv zobrazuje i stav automatického režimu. Jedná se o: *Připraveno ke spuštění*, *Automatický režim se spouští*, *Automatický režim v chodu*, *Automatický režim se zastavuje*, *Automatický režim zastaven*.



Obrázek 34: Navigační menu automatického režimu

#### Legenda

1. Tlačítko pro vyčištění tabulky
2. Tlačítko pro reset tabulky
3. Softwarové tlačítko Zap/Vyp
4. Tlačítko na další stránku tabulky
5. Tlačítko pro načtení a uložení tabulky
6. Informační lišta

Popis navigačních tlačítek na stránce automatického režimu, zobrazených na obrázku (Obrázek 34):

- ***Vyčistit tabulku***

Tlačítko pro vyčištění tabulky zastaví všechny běžící uzly kvůli zamezení nastání chyby a poté nastaví tabulku a automatický režim do výchozího nastavení. Výchozí nastavení odpovídá prvotnímu spuštění vizualizace (viz Obrázek 29).

- **Reset tabulky**

Tlačítko pro resetování tabulky nastaví automatický režim do výchozího nastavení, avšak ponechá naplněnou tabulku se všemi akcemi a podmínkami. Pro případ když uživatel potřebuje spustit celý výrobní proces znovu od začátku.

- **Softwarové tlačítko Zap/Vyp**

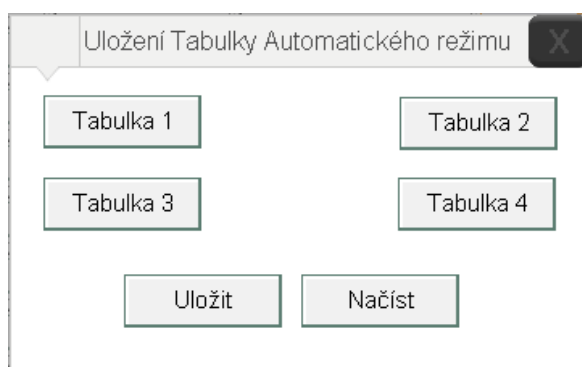
Tlačítko slouží jako pomocné, při podmínce přechodu na tlačítko. Například v případě že obsluha chce přejít na další akční člen až v případě, kdy vizuálně zhodnotí situaci stroje a nechce používat hardwarová tlačítka na ovládacím panelu. Tlačítko se nachází uprostřed navigační lišty.

- **Tlačítko na další stránku tabulky**

Po stisknutí tlačítka se zobrazí další stránka tabulky. Na první stránce jsou řádky 1-16 a na druhé 17-32. Celkem je tedy možné naplnit tabulku o 32 řádcích.

- **Načíst/Uložit tabulku**

Po stisknutí tlačítka pro načtení a uložení tabulky vyskočí okno, kde jsou na výběr čtyři předvolby (viz Obrázek 35). Po stisknutí jedné z předvoleb a následného stisknutí tlačítka „uložit“ se provede nahrání aktuální uživatelem vyplněné tabulky do dané předvolby pro pozdější použití. V opačném případě, po zvolení předvolby a stisknutí tlačítka „Nahrát“, se již uložená tabulka v předvolbě nahraje do tabulky aktuální.



Obrázek 35: Okno pro načtení a uložení tabulky

## Závěr

Cílem diplomové práce bylo seznámení autora s aktuálním mechanickým, elektrickým a programovým řešením experimentální linky na výrobu nanovlákných produktů. Seznámení se s ústrojím experimentální linky bylo splněno v kapitole 3, kde autor poskytuje popis komorového přízového spinneru. Popis a vysvětlení jednotlivých uzlů mechanického ústrojí linky se nachází v kapitole 4. Na základě těchto poznatků o mechanickém uspořádání byl autorem vytvořen algoritmus založený na předpokladech obsluhy pro snadnou změnu konfigurace. Algoritmus byl v průběhu vytváření programového bloku několikrát pozměněn tak, aby bylo možné dodržet předpoklady pro řízení a bylo možné následně vytvořit požadované grafické rozhraní.

V kapitole 5 bylo přistoupeno k samotnému návrhu softwaru pro řízení linky. Z vycházejícího návrhu a vývojových diagramů byl vytvořen programový blok. Program je vytvořený ze dvou složených stavových automatů. První automat zajišťuje kontrolu vnějších faktorů, bez kterých nemůže dojít ke spuštění linky. Po spuštění stavový automat zajišťuje rozpoznání, v jakém režimu se automat nachází. Druhý automat, který je vnořený do prvního, zajišťuje výběr mechanických částí a to tak, aby byl dodržen uživatelem navolený program na konkrétní výrobu nanopříze. Po načtení navolené produkce z grafického uživatelského rozhraní a následnému zapsání do tabulky, program projde tabulku a podle zvoleného akčního členu vykoná požadované akce. Mezi další části programu patří možnost uložení aktuální uživatelem vyplněné tabulky do dané předvolby pro pozdější použití. Resetování tabulky slouží k možnosti vykonání aktuálního navoleného programu od začátku, například kdyby došlo k nečekané poruše a obsluha musela výrobní linku zastavit. Vycištění tabulky nastaví tabulku a automatický režim do výchozího nastavení, aby mohla obsluha navolit novou konfiguraci výroby. Vytvořený programový blok a jeho dílčí části jsou popsány v kapitole 5.4 tak, aby autor čtenáři vysvětlil, jaké funkce jsou zde použity.

V další kapitole autor ukazuje, jakým způsobem bylo vytvořeno grafické uživatelské rozhraní a jakým způsobem je propojeno s vytvořeným programovým blokem. Jsou zde popsány části vizualizace a automatického režimu, aby obsluha získala kompletní přehled o funkčnosti stroje a nastavitelnosti požadované výroby.

V poslední kapitole byl autorem vytvořen návod na použití vytvořené vizualizace, aby obsluha mohla snadno nastavovat a modifikovat parametry produkce v automatickém režimu. Jsou zde popsány metody zadávání výrobního procesu do



tabulky, a jakým způsobem je možné tabulku ovládat, pro docílení maximální efektivity výroby.

Na závěr byl vytvořený program a grafická vizualizace odzkoušena na reálném mechanickém ústrojí, které se nachází v laboratoři Katedry netkaných textilií a nanovlákněných struktur na Technické Univerzitě v Liberci.

## Seznam použité literatury

- [1] B&R AUTOMATION. [online]. [cit. 2017-05-15]. Dostupné z: <https://www.br-automation.com/cs/perfection-in-automation/>
- [2] ŠMEJKAL, L.; MARTINÁSKOVÁ M. 1999. PLC a automatizace. 1. vyd. Praha: BEN – technická literatura. ISBN 80-860-5658-9.
- [3] X20 Systems. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/cs/produkty/io-systems/x20-system/>.
- [4] X67 Systems. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/cs/produkty/io-systems/x67-system/>.
- [5] Ethernet Powerlink – komunikace v reálném čase. *Automa: časopis pro automatizační techniku* [online]. [cit. 2017-05-15]. Dostupné z: [http://automa.cz/cz/casopis-clanky/ethernet-powerlink-komunikace-v-realnem-case-2001\\_03\\_33500\\_2769/](http://automa.cz/cz/casopis-clanky/ethernet-powerlink-komunikace-v-realnem-case-2001_03_33500_2769/).
- [6] BÍLEK, K. Produkty B&R – automatizace která má vtip. *Automa: časopis pro automatizační techniku* [online]. 2005 [cit. 2018-04-02]. Dostupné z: [http://automa.cz/cz/casopis-clanky/produkty-b-r-automatizace-ktera-ma-vtip-2005\\_03\\_30375\\_2006/](http://automa.cz/cz/casopis-clanky/produkty-b-r-automatizace-ktera-ma-vtip-2005_03_30375_2006/).
- [7] CANopen. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/en/products/networks-and-fieldbus-modules/canopen/>.
- [8] JOHN, KARL-Heinz; TIEGELKAMP, Michael. IEC 61131-3: programming industrial automation systems : *Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making aids. 2nd ed.*
- [9] ŠMEJKAL, Ladislav. Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 (část 4). *Automa: časopis pro automatizační techniku*[online]. [cit. 2017-05-15]. Dostupné z: [http://automa.cz/cz/casopis-clanky/esperanto-programatoru-plc-programovani-podle-normy-iec/en-61131-3-cast-4-2011\\_12\\_45413\\_5828/](http://automa.cz/cz/casopis-clanky/esperanto-programatoru-plc-programovani-podle-normy-iec/en-61131-3-cast-4-2011_12_45413_5828/).
- [10] Jan Valtera. „Obecný popis AC zvláknovacího zařízení KOPRIS 2“. Liberec,2018.

- [11] Přístrojový transformátor napětí podpěrný VTS 38. *KPB INTRA s.r.o.* [online].  
[cit. 2018-11-04]. Dostupné z: <https://www.kpb intra.cz/produkty/vts-38>