



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

NÁVRH ŘÍDICÍHO PLC ROBOTIZOVANÉ VÝROBNÍ BUŇKY

DESIGN OF THE PLC PROGRAM FOR ROBOTIC WORK CELL CONTROLLING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Štěpán Žalud

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jakub Bražina

BRNO 2021

Zadání bakalářské práce

Ústav:	Ústav výrobních strojů, systémů a robotiky
Student:	Štěpán Žalud
Studijní program:	Strojírenství
Studijní obor:	Základy strojního inženýrství
Vedoucí práce:	Ing. Jakub Bražina
Akademický rok:	2020/21

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Návrh řídicího PLC robotizované výrobní buňky

Stručná charakteristika problematiky úkolu:

Práce je zaměřená na tvorbu PLC programu pro řízení zadané buňky. Buňka se skládá z obráběcího stroje, stacionární brusky, zakládací a odkládací stanice, potřebné senzory a bezpečnostních prvků. Navržený PLC program, včetně vizualizace, bude sloužit jako demonstrační úloha ve výuce. V průběhu tvorby této práce se student seznámí s problematikou navrhování a tvorby řídicích PLC programů, včetně jejich vizualizace.

Cíle bakalářské práce:

Rešerše dané problematiky.
Seznámení s buňkou a volba jednotlivých periférií.
Návrh řídicího PLC programu.
Návrh HMI.
Vlastní závěry a doporučení pro praxi.

Seznam doporučené literatury:

ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace 1: 1. základní pojmy, úvod do programování. Praha: BEN, 1999. ISBN 80-86056-58-9.

MARTÍNEK, Radislav. Senzory v průmyslové praxi. Praha: BEN, 2004. ISBN 80-7300-114-4.

Beckhoff Information System [online]. Verl: Beckhoff Automation GmbH & Co., 2020 [cit. 2020-11-03]. Dostupné z: https://infosys.beckhoff.com/index_en.htm

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Assignment Bachelor's Thesis

Institut: Institute of Production Machines, Systems and Robotics
Student: **Štěpán Žalud**
Degree programm: Engineering
Branch: Fundamentals of Mechanical Engineering
Supervisor: **Ing. Jakub Bražina**
Academic year: 2020/21

As provided for by the Act No. 111/98 Coll. on higher education institutions and the BUT Study and Examination Regulations, the director of the Institute hereby assigns the following topic of Bachelor's Thesis:

Design of the PLC program for robotic work cell controlling

Brief Description:

The thesis is focused on the PLC program creation to control given work cell. The work cell consists of the machine tool, stationary grinding machine, loading and storage station, necessary sensors and safety elements. Designed PLC program will be used as a sample task during tuition. During work on this thesis, a student will be acquainted with problematics of the design and creation of the controlling PLC programs, including its visualization.

Bachelor's Thesis goals:

Research of the issue.
Acquaintance with the work cell and peripherals choice.
Controlling PLC program design.
HMI design.
Conclusions and recommendations for the practice.

Recommended bibliography:

ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace 1: 1. základní pojmy, úvod do programování. Praha: BEN, 1999. ISBN 80-86056-58-9.

MARTÍNEK, Radislav. Senzory v průmyslové praxi. Praha: BEN, 2004. ISBN 80-7300-114-4.

Beckhoff Information System [online]. Verl: Beckhoff Automation GmbH & Co., 2020 [cit. 2020-11-03]. Dostupné z: https://infosys.beckhoff.com/index_en.htm

Deadline for submission Bachelor's Thesis is given by the Schedule of the Academic year 2020/21

In Brno,

L. S.

doc. Ing. Petr Blecha, Ph.D.
Director of the Institute

doc. Ing. Jaroslav Katolický, Ph.D.
FME dean

ABSTRAKT

Cílem této bakalářské práce je sestavení řídicího PLC programu robotizované výrobní buňky spolu s návrhem a vizualizací HMI panelu. Teoretická část pojednává o problematice týkající se automatizace výrobních buněk a začlenění PLC automatů. V praktické části dochází k seznámení se zvolenou strukturou programu a jejími klíčovými prvky a následně představení vizualizace HMI panelu. Práce je doplněna o soubor obsahující sestavený kód a video demonstrující funkci programu.

ABSTRACT

The aim of this bachelor's thesis is to compile a PLC control program of a robotic work cell together with the design and visualization of an HMI panel. The theoretical part deals with issues related to the automation of production cells and the integration of PLCs. In the practical part, get acquainted with the selected structure of the program and its key elements and the following presented visualization of the HMI panel. The work is supplemented by a file containing the compiled code and video demonstration function of the program.

KLÍČOVÁ SLOVA

PLC, programovatelné logické automaty, automatizace, HMI, řídicí panel, výrobní buňka, robotizovaná buňka, robot

KEYWORDS

PLC, programmable logic controller, automation, HMI, Human machine interface, work cell, robotic cell, robot

BIBLIOGRAFICKÁ CITACE

ŽALUD, Š. Návrh řídicího PLC robotizované výrobní buňky, Brno, Vysoké učení technické v Brně, Fakulta strojního inženýrství. 2021, 48 s., Vedoucí bakalářské práce Ing. Jakub Bražina.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu mé bakalářské práce, panu Ing. Jakubu Bražinovi za vstřícný přístup a vedení během konzultací a v průběhu tvorby celé této práce.

Žalud Štěpán

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Jakuba Bražiny a s použitím literatury uvedené v seznamu.

V Brně dne 21. května 2021

.....

Žalud Štěpán

OBSAH

1	ÚVOD.....	3
2	MOTIVACE.....	5
3	přehled současného stavu poznání.....	7
3.1	Automatizovaná výroba	7
3.2	Virtuální zprovoznění.....	7
3.3	Programmable logic controlers	8
3.3.1	Typy PLC	10
3.3.2	Software.....	10
3.4	HMI	11
3.5	Senzorika.....	12
3.6	Bezpečnost	13
3.6.1	Hardwarové bezpečnostní prvky	14
3.6.2	Softwarové bezpečnostní prvky	14
3.6.3	Bezpečnostní PLC	15
4	Praktická část.....	19
4.1	Zadané pracoviště.....	19
4.1.1	Model zadané výrobní buňky a pracovní cyklus.....	19
4.1.2	Zpracovávané signály	20
4.2	PLC program.....	22
4.2.1	Struktura řídicího kódu.....	22
4.2.2	FB_HMI_Automode.....	24
4.2.3	FB_ProductionRoutine	27
4.2.4	FB_RobotProgram.....	29
4.2.5	FB_MachineCenter.....	32
4.2.6	FB_Conditions.....	35
4.2.7	FC_Grinder.....	38
4.2.8	FC_VariablesAssignment	38
4.2.9	PLC tagy.....	39
4.3	HMI panel	40
4.3.1	Vizualizace	41
4.3.2	Alarmy a texty	42
4.3.3	Pop-up screens.....	46
4.4	Virtuální zprovoznění.....	46

5	Zhodnocení a diskuze.....	49
6	Závěr.....	51
7	Seznam použitých zdrojů	53
8	Seznam zkratk, symbolů, obrázků a tabulek	55
8.1	Seznam tabulek	55
8.2	Seznam obrázků	55
8.3	Seznam zkratk	56
9	Seznam příloh.....	57

1 ÚVOD

V současné době, kdy neustále narůstající poptávka po produktech průmyslové výroby klade stále se zvyšující nároky na kvantitu při zachování, či dokonce navýšení kvality, je každá větší firma nucena dříve či později začlenit do výroby automatizované výrobní celky. Ty za cenu vyšších pořizovacích nákladů poskytují řadu výhod, mezi které můžeme počítat předvídatelnost, přesnost, zachování kvality, efektivitu a mnohdy nepřetržitou výrobu.

Uvedení takového pracoviště do provozu však představuje komplexní úkon, během kterého dochází k návrhu, tvorbě a následně testování vzájemně se doplňujících částí. Nedílnou součástí tak představuje příprava řídicí soustavy, která je mozkiem automatizovaných výrobních buněk. K těmto účelům může sloužit, mimo jiné, právě programovatelný logický automat neboli PLC, které představuje vhodnou variantu do poměrně náročných podmínek výrobních hal.

Tato práce se zabývá návrhem kódu, kterým bude prostřednictvím PLC řízen jednoduchý model robotizované výrobní buňky. Současně bude navržen a vizualizován HMI panel, pomocí kterého bude celý proces říze

2 MOTIVACE

Automatizace a virtuální zprovoznění je současnost a bezpochyby i budoucnost nejen průmyslu. Návrh řídicích prvků těchto soustav tak představuje perspektivní obor, co se profesního uplatnění týče. Tato práce představuje vhodný způsob jak se seznámit s problematikou automatizovaného řízení, základními pojmy a principy, které je nezbytné si pro orientaci v této oblasti osvojit.

Cílem práce bylo navrhnout funkční kód a řídicí panel jednoduchého modelu výrobní buňky a získání základních teoretických i praktických znalostí, které poskytnou solidní základ pro další rozvoj vědomostí na poli automatizovaného řízení.

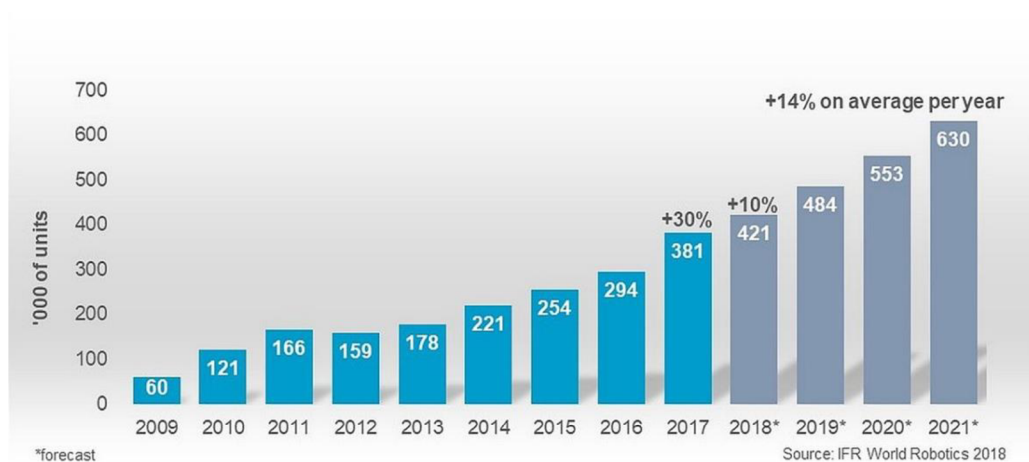
3 PŘEHLED SOUČASNÉHO STAVU POZNÁNÍ

3.1 Automatizovaná výroba

S rychlostí rozvoje technologií a digitalizace v průběhu 21. století, začaly úměrně stoupat nároky zákazníků na kvalitu, cenu i rychlost dodání požadovaného produktu. Tím je vyvíjen značný tlak na výrobce, kteří postupně začínají být nuceni k zavedení automatizovaných procesů, ať už v menší či větší míře. Na obrázku 3-1 je vidět počet světově dodaných průmyslových robotů do roku 2017 a tehdejší odhad pro roky následující.

Automatizované výrobní procesy, za cenu vyšších pořizovacích nákladů na hardware i software, nabízí majitelům výrobních linek výhody jednoznačně přispívající postupnému nahrazování lidské pracovní síly. Patří mezi ně zejména stabilní produkce, vyšší efektivita výroby a v neposlední řadě vyšší zisk v dlouhodobém měřítku. Tyto faktory jsou zajištěny zejména absencí aktivit, nezbytných pro lidskou obsluhu, jako jsou pauzy na oběd, odchod na WC a podobně.

Vzhledem k tomu, že se zpravidla jedná o uskupení vzájemně kolaborujícího hardwaru a softwaru, je uvedení takovéto pracovní buňky do provozu poměrně složitá a časově náročná operace. Tento fakt jde v přímém rozporu s nároky na vysokou produktivitu výrobních linek, kdy si firma nemůže dovolit odstávku v řádech týdnů, či dokonce měsíců, z důvodů zavedení automatizovaných procesů. Zde přichází na scénu tzv. virtuální zprovoznění.



3-1) Počet světově dodaných robotů do roku 2017

3.2 Virtuální zprovoznění

Jako virtuální zprovoznění lze označit proces, při kterém je v prostředí počítačového softwaru zkonstruován model výrobní linky totožný se svým fyzickým protějškem. Model musí dosahovat úrovně, kdy věrně simuluje rozměry, kinematiku i komunikaci jednotlivých periférií. Tento model následně slouží k ověření funkčnosti před samotnou montáží. Virtuální zprovoznění tak umožní optimalizaci i kontrolu funkčnosti a bezpečnost ve stádiu vývoje. Dojde-li tak k nálezů závady, je možné ji odstranit s minimálními vedlejšími dopady, mezi které by za normálních okolností patřily zejména finanční ztráta, poškození jednotlivých periférií nebo ohrožení bezpečnosti lidské obsluhy. Dále lze mezi výhody zařadit nezávislost celého tohoto procesu na dalších úkonech, jako je například fyzická montáž linky. Vše potřebné se

tedy může začít chystat s dostatečným časovým předstihem, ať už se jedná o zavedení celé nové výrobní jednotky, nebo pouze o změnu zaměření, jednotky již existující. Samotná odstávka nebo montáž je tedy omezena na nezbytné minimum, během které dojde k sestavení hardwaru a nahrání již vyvinutého a ověřeného softwaru [1]

Co tedy obecně přináší virtuální zprovoznění[1]:

- úsporu času, rychlejší uvedení stroje do provozu;
- nižší náklady, snížení rizika chyb;
- pružnější výrobu, snazší provádění změn;
- efektivnější práci, technici mohou pracovat paralelně;
- vyšší kvalitu.

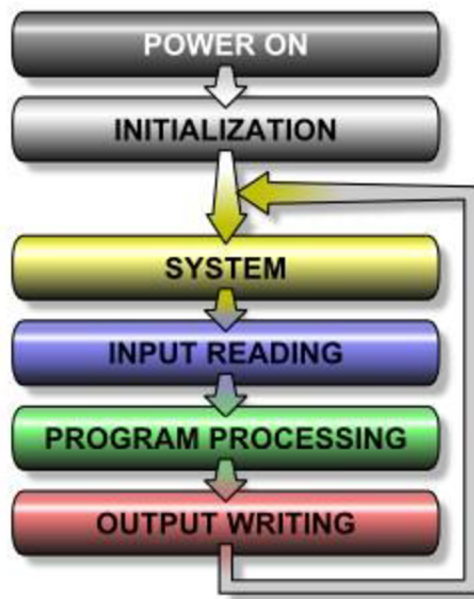
Nedílnou součástí celého procesu je rovněž příprava řídicího PLC .

3.3 Programmable logic controllers

Programmable logic controller, česky programovatelný logický automat, dále už jen PLC, je jednotka zodpovídající za řízení jednotlivých periférií automatizované výrobní buňky.

S prvními druhy PLC se bylo možné setkat už počátkem 70. let minulého století [2]. Fakt, že se takový typ zařízení udržel ve světě techniky po tak dlouhou dobu, jednoznačně potvrzuje jeho nepostradatelnou funkci v automatizovaných procesech. Zjednodušeně se dá popsat 3 částmi: vstupy, výstupy a řídicím CPU. Vstupy i výstupy mohou zpracovávat signály jak digitální, pracující na principu vypnuto/zapnuto tak analogové, které se dají využít pro regulaci nebo snímání tlaku, otáček, teploty a dalších. Dnešní PLC jsou schopny běžně zpracovávat 1000 až 10000 signálů [3].

Princip funkce je poměrně jednoduchý (Obr. 3-2). Po inicializaci a kontrole systému jsou na vstupy přivedeny signály, které jsou přečteny, a dočasně uloženy do paměti CPU. Ten následně na základě nahraného programu, nastaví výstupy. Celý tento proces se periodicky opakuje. Zásadní funkcí PLC je, že celý tento proces probíhá v reálném čase. V závislosti na programu a počtu a typu operací, které musí PLC vykonat, je i doba cyklu proměnlivá. Jako dobu jednoho cyklu můžeme definovat čas, za jak dlouho PLC vykoná všechny potřebné úlohy a navrátí se do výchozího pevně definovaného bodu [4] [5].



3-2) Cyklus PLC [5]

Výhody tohoto systému se dají shrnout následovně [6]:

- Spolehlivost – oproti běžným počítačům, které jsou určeny pro práci v laboratorních a kancelářských podmínkách, zvládají logické automaty pracovat s vynikající spolehlivostí i v prašném prostředí výrobních hal [8].
- Univerzálnost – Není potřeba vyvíjet nové systémy pro nové procesy, stačí pouze vybrat a vhodně sestavit již existující hardware [8].
- Snadná obsluha – Díky svým vlastnostem a možnosti připojení HMI panelu, kde jsou odfiltrovány nevyužité funkce PLC, je tato řídicí jednotka poměrně snadná a nenáročná na obsluhu.
- Programování a následné úpravy – Díky programovacím softwarům dodávaným s PLC není příprava řídicího programu příliš náročná. Zároveň je dobře umožněno provádět další úpravy programu, v případě zjištěných nedostatků, nebo úpravy zadání.

Nelze však tvrdit, že tento systém je jednoznačně nejlepší volbou pro všechny případy řízení. Ostatně i u PLC lze najít určité nevýhody, které musí být uvažovány [6]:

- Delší doba trvání cyklu – Vzhledem k použití předpřipravených softwarů nemusí být vždy výsledný kód nejefektivnější. Výsledná doba cyklu se tak běžně pohybuje v řádech desítek až stovek milisekund [8].
- Vysoká cena vzhledem k výkonu
- Nekompatibilita periférií a aplikací napříč různými výrobci
- Zbytečné funkce – vzhledem k univerzálnosti programovatelných automatů, se může stát, že v dodávce jsou zaplacené funkce, které zákazník pro svou potřebu nevyužije

PLC můžeme najít v portfoliu mnoha firem. Mezi nejrozšířenější patří Siemens, Rockwell Automation, Mitsubishi Electric Corporation, Schneider Electric, Omron Corporation atd. [7] Nejde jednoznačně určit kvalitnějšího výrobce. Vždy se musí zvážit několik faktorů (cenová nabídka, využití...) a tudíž je i volba výrobce individuální.

3.3.1 Typy PLC

V zásadě lze rozlišovat mezi 3 základními typy PLC. Těmi jsou PLC kompaktní, PLC modulární a PLC bezpečnostní. V této kapitole budu představeny 2 zmíněné typy [9].

Bezpečnostním PLC je věnována samostatná kapitola 1.6.3 Bezpečnostní PLC.

Kompaktní PLC

Je určeno především pro menší a jednoduché systémy. V jednom zařízení tak lze nalézt kombinaci všech základních prvků potřebných pro řízení procesu. Krom řídicího CPU tak kompaktní PLC obsahuje jak digitální, tak analogové vstupy a výstupy. Poměr vstupů vůči výstupům se obecně pohybuje v poměru 2/1 u digitálních a 4/1 u analogových signálů. Krom nižší ceny je výhodou kompaktních PLC také rychlost zpracování dat. To je způsobeno absencí řadiče sběrnice. Díky tomu je dosaženo doby trvání cyklu řádově v jednotkách milisekund.

Modulární PLC

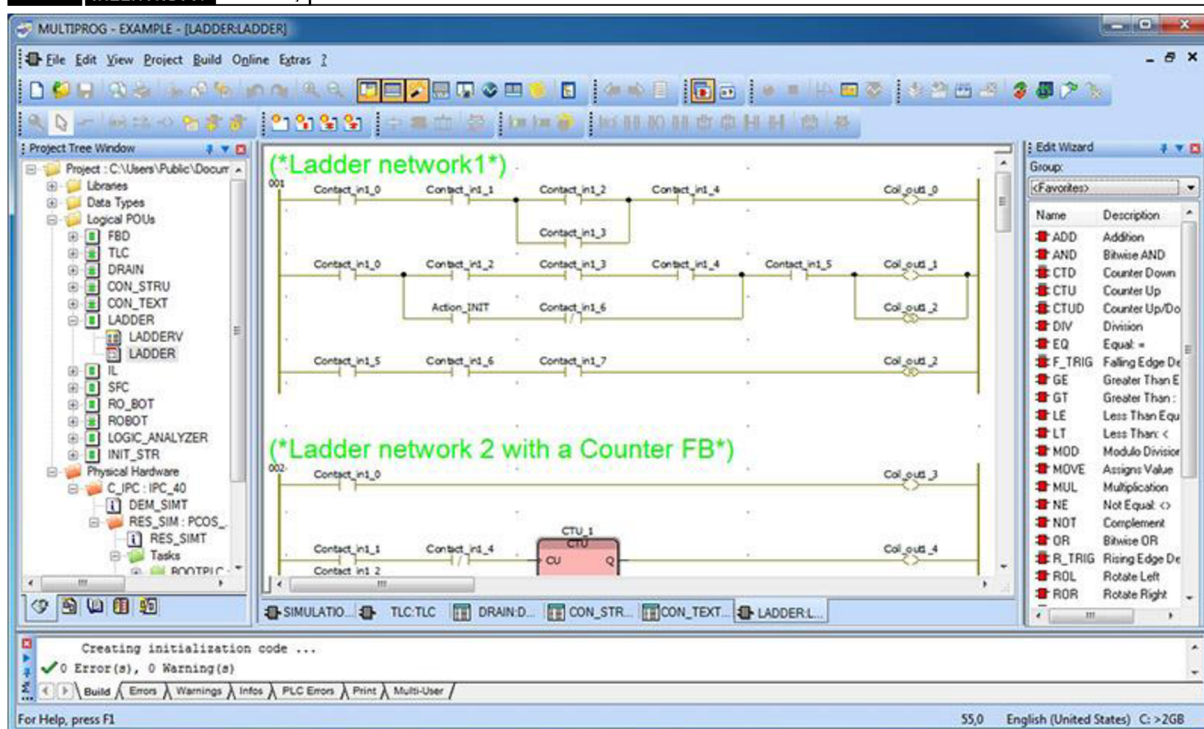
Jak už název napovídá, představuje modulární PLC řešení pro potřeby řízení složitějších procesů. Na základě požadavků je řídicí jednotka sestavena z potřebných modulů. Těmi jsou centrální CPU a další periferní moduly pro vstupy, výstupy, komunikaci a další. Vzhledem k předpokládanému zpracování většího počtu signálů je CPU vybaveno větší operační pamětí než dříve zmíněné kompaktní PLC. Ve prospěch modulárních PLC hraje hlavně flexibilita a modularita. Jejich hlavní nevýhodou je pak vyšší pořizovací cena a větší složitost jednotky jako takové.

3.3.2 Software

Programování PLC obecně nabízí poměrnou variabilitu v tom, jakým způsobem se bude řídicí kód sestavovat. Dle normy ČSN EN 61131-3 ed. 2 je definováno 5 programovacích jazyků [10].

- Jazyk strukturovaného textu (ST)
- Jazyk funkčního blokového schématu (FBD)
- Sekvenční funkční diagram (SFC)
- Jazyk seznamu instrukcí (IL)
- Jazyk příčkového diagramu (LD)

Co se použití týče, je nejrozšířenější právě Jazyk příčkového diagramu (Obr. 3-3). [10] Bohužel není vždy u všech PLC dostupný každý programovací jazyk. Rovněž nebývá software kompatibilní napříč jednotlivými výrobci. Nemusí tak být dokonce ani u jednoho výrobce v rámci různých řad výrobků.



3-3) Jazyk příčkového diagramu [10]

Volba programovacích jazyků je tak ovlivněna možnostmi zvoleného PLC, preferencí programátora a určením celé sestavy. Samotnou volbu je dobré předem patřičně zvážit. V pozdější fázi může zásadně ovlivnit případnou údržbu nebo aktualizaci řídicího kódu.

Jak již bylo zmíněno PLC systémy různých výrobců nejsou kompatibilní. Jednotlivé značky mají tedy své vlastní softwary určené pro tvorbu řídicích PLC programů neboli IDE (integrated development environment). Jako příklad uvedu programy 2 nejrozšířenějších firem [10].

Tia portal – Produkt značky Siemens, který sdružuje vše potřebné pro automatizované procesy. Mimo jiné tak umožňuje programování řídicích PLC kódů ve fázi vývoje i implementace do výrobního procesu. Zároveň TIA portal umožňuje přípravu a vizualizaci HMI panelů

Studio 5000 – Zástupce firmy Rockwell automation, hojně rozšířený zejména v Americe, nabízí možnost programovat ve všech 5 programovacích jazycích a libovolně mezi nimi přepínat.

Ač si mohou být programovací softwary v určitých ohledech podobné, znalost jednoho programu nezaručuje bezproblémové zvládnutí programu jiného výrobce, ačkoliv dává dobrý předpoklad k osvojení si potřebných znalostí.

3.4 HMI

HMI neboli Human-Machine Interface je, jak už název napovídá, rozhraní mezi lidskou obsluhou a řízeným strojem (Obr. 3-4). Mohlo by se zdát, že se tedy jedná o řídicí prvek daného procesu. Tato domněnka by však byla mylná. Jedná se pouze o prostředek, který umožňuje lidské obsluze komunikovat s řídicím prvkem, v našem případě s PLC. PLC totiž v základu

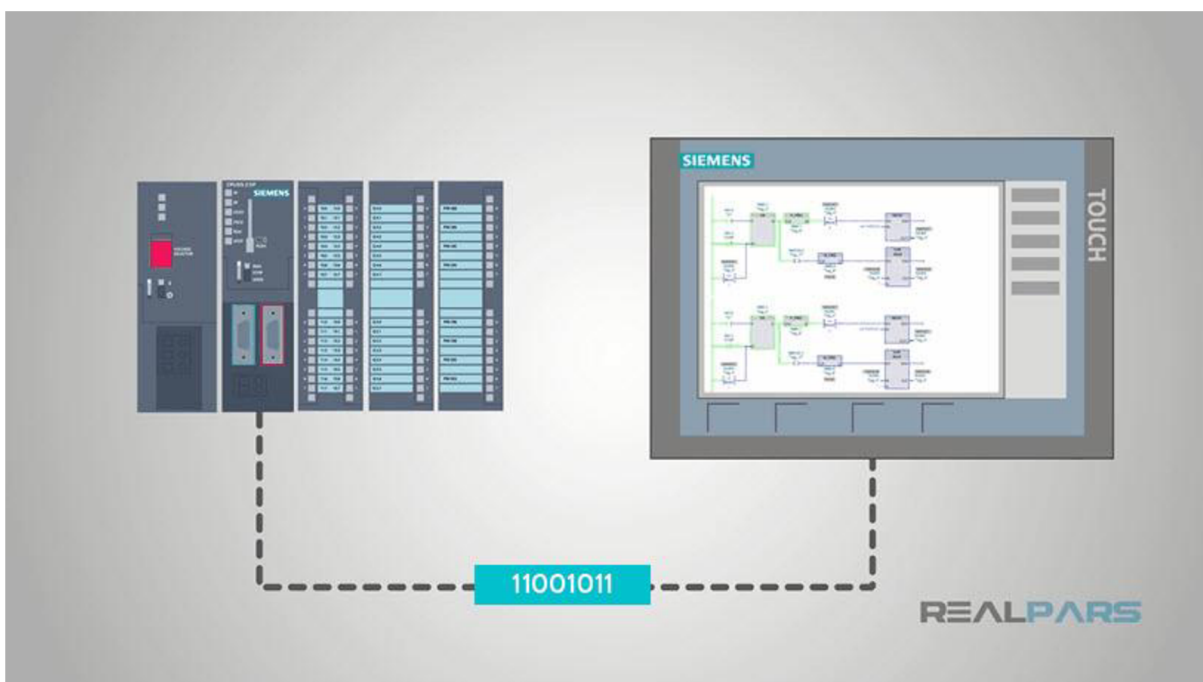
neumožňuje pozorovat probíhající proces, aniž by byl externě připojen k další periférii, například právě k HMI panelu. Díky HMI je možné získávat informace z námi pozorovaného procesu ve formě textu nebo grafické vizualizace.

HMI tak nabízí další zjednodušení pro obsluhu, která se tak nemusí vyznávat v samotném programu nahraném do PLC, ale může názorně pozorovat dění na tomto panelu. V našem případě by rovněž měla HMI vizualizace sloužit jako verifikace funkčnosti sestaveného řídicího programu.

V zásadě se dá HMI rozdělit na aktivní a pasivní panely. Zatímco u pasivních panelů probíhá komunikace jednosměrně, aktivní panely mohou fungovat i na principu otázka – odpověď. Mluvíme tedy o dvousměrné komunikaci.

V praxi se lze setkat s různými konstrukčními řešeními. Můžeme tedy narazit na dotykové obrazovky, klávesnice, numerické klávesnice, tlačítka, joysticky, různé porty (USB, ethernet) a další. Nutno podotknout, že úměrně s komplikovaností panelu může klesat jeho spolehlivost. Je tedy vždy nutné zvážit jakou konfiguraci a rozlišení nezbytně vyžadujeme pro naše potřeby [11] [12].

Jelikož použití HMI panelů úzce souvisí s PLC, není divu, že produkcí se zabývají již zmíněné firmy jako Siemens, Schneider Electric, Omron atd.



3-4) Vizualizace komunikace HMI s PLC

3.5 Senzorika

Aby bylo vůbec možné dovést proces automatizace ke zdárnému konci, je nezbytné doplnit celý systém o řadu senzorů. Ty mohou mít bezpečnostní tak informativní funkci. Poskytují řídicímu prvku nezbytné informace o poloze a stavu periférií výrobní buňky, jejím prostředí nebo o součástech se kterými je v rámci výrobního procesu manipulováno. V praxi se můžeme setkat i s označením jako čidlo, detektor, snímač a další. Vždy se však jedná o stejný typ zařízení.

Senzory můžeme dělit následovně [14]:

- **Fyzikální princip** – odporové, indukční, kapacitní, magnetické, optické, a další.
- **Dotykové a bezdotykové**
- **Analogové a diskrétní**
- **Typ měřené veličiny** – tlakové, tepelné, radiační...
- **Výrobní technologie** – elektromechanické, elektrické, pneumatické, polovodičové

Pomineme-li bezpečnostní senzory, kterým bude věnována pozornost v kapitole 1.6 Bezpečnost, setkáme se v této práci zejména se sensorikou týkající se přítomnosti součástí v zakládací a odkládací stanici, obráběcím centru a dalších.

Je několik typů senzorů, které mohou být pro podobné účely použity. Konkrétní typ použitého čidla by se mimo jiné vybíral i na základě materiálu, se kterým bude manipulováno. Protože tato informace není pro zadání úlohy blíže specifikována, bude uvedeno několik typů senzorů, které by tuto funkci mohli plnit. Pro účely napsání PLC programu však není nutné konkrétní typ znát, protože získávaná informace se neliší. Jde pouze o údaj, zda senzor detekuje přítomnost předmětu či nikoliv.

Příklady typů senzorů [13]:

Elektro-mechanické – Základní typ senzoru, který změni svůj stav ve chvíli, kdy vlivem fyzického kontaktu s detekovanou součástí dojde k vychýlení jeho stykové části, v podobě páčky, rotujícího válečku a dalších. Jeho nevýhodou je potřeba bezprostředního kontaktu, který není vždy možný nebo vhodný.

Indukční senzor – Tento senzor detekuje narušení elektromagnetického pole vytvářeného tímto snímačem. Je schopen bezkontaktního snímání až na několik desítek mm. Díky svým vlastnostem a výborné spolehlivosti i v prašném prostředí, je hojně využíván v průmyslových výrobních halách. Za předpokladu, že by v naší výrobní buňce bylo manipulováno s kovovým, vodivým materiálem, s největší pravděpodobností by bylo využito právě Indukčních senzorů.

Kapacitní: Princip jejich funkce je velmi podobný jako u senzorů indukčních. Místo detekce elektromagnetického pole však reagují na změnu pole elektrostatického. Tato vlastnost je předurčuje jako vhodné senzory při práci s nekovovými materiály jako jsou dřevo, papír, plast a další.

Magnetické – Magnetické senzory se používají pro detekci magnetických předmětů a reagují tak na změnu magnetického pole. Jeho výhodou je zejména prostupnost i skrz nemagnetické materiály a oproti indukčním senzorům větší dosah.

Zmíněné typy senzorů slouží jen jako ukázka některých typů a principů funkce.

V praxi je však možné se setkat s dalšími senzory, například optickými, ultrazvukovými atd.

3.6 Bezpečnost

S technologickým vývojem, který průmysl v posledních letech zažívá, značně stoupají rizika plynoucí z případných poruch, chybné obsluhy, havárií či dokonce zneužití komplikované

techniky. Z toho důvodu je nezbytné do plánu přípravy jakéhokoliv pracoviště, ať už automatizovaného či nikoliv, zahrnout i zhodnocení případných bezpečnostních rizik a posléze přijetí náležitých opatření a jejich implementace do procesu vývoje. Cílem takového zhodnocení je předpovězení a následně předejití všech možných scénářů, kdy dojde ke kolizi jednotlivých periférií navzájem nebo s lidskou obsluhou. Případná kolize může znamenat významnou finanční ztrátu vlivem poškození strojů nebo prostým odstavením linky. Opomenout se nesmí ani případné riziko zdravotní újmy nebo dokonce smrt obsluhy.

Celkové zhodnocení a navržení bezpečnosti podobného pracoviště, jako je předmětem této práce, by však vydalo na samostatnou práci nemalého rozsahu. Z toho důvodu budou dále uvažovány hlavně bezpečnostní prvky, které ovlivňují řídicí PLC program a bezprostředně v něm mohou vystupovat.

3.6.1 Hardwarové bezpečnostní prvky

Jedná se o fyzické periferie, které mohou mít za úkol zamezit lidské obsluze vstup na pracoviště během provozu, nebo nouzové zastavení linky v případě zjištěné poruchy. Právě tyto prvky je nutné mít jasně stanovené před samotným vývojem PLC programu. Zásadním způsobem tak ovlivní finální podobu řídicího kódu. Prodejem podobných bezpečnostních prvků se zabývá řada známých firem. Jako příklad lze uvést firmy ABB nebo Schneider Electric

Dále je uvedeno několik příkladů hardwarových bezpečnostních prvků, se kterými se můžeme na podobných pracovištích setkat.

Bezpečnostní hrazení a oplocení

Jedná se o fyzickou zábranu ohraničující prostor výrobní buňky, zamezující vniknutí z jiných než jasně stanovených a zabezpečených míst, jako jsou okna, dveře a další manipulační otvory. Hrazení samo o sobě na rozdíl od snímačů u dveří atd., neposkytuje pro řídicí systém žádné informace o svém stavu.

Bezpečnostní zámek

U výrobních buněk se s ním běžně setkáme u dveří v ochranném hrazení tak, aby zamezil vstupu v době, kdy je linka v běhu, popřípadě ji okamžitě odstavil z provozu.

Světelné závory

Zamezují narušení procesu v místech, kde je vhodné ponechat stále otevřený prostor, například z důvodu dopravníku zásobujícího výrobní buňku. Jejich výhodou je poměrně velký dosah, až několik desítek metrů, čímž jsou schopné zabezpečit i velké otvory v hrazení pracovního prostoru. Hlídaný prostor chrání pomocí sítě paprsků, při jejichž přerušení dojde k zastavení stroje.

Nášlapné rohože

Své uplatnění naleznou zejména při detekci polohy lidské osoby. Jejich výhodou je možnost snímání relativně velké plochy. Fungují na principu spojení dvou kontaktních ploch při zatížení rohože.

3.6.2 Softwarové bezpečnostní prvky

Mezi softwarové bezpečnostní prvky se řadí programy hlídající bezpečnost řízených strojů. Může se tak jednat například o software, který je obsažen v programu robotu. Ten snímá jeho polohu a hlídá, aby se robot svými koncovými body nedostal mimo jemu vyhraněný pracovní prostor [15].

3.6.3 Bezpečnostní PLC

Je-li pozornost věnována bezpečnosti týkající se prostředí automatizace a PLC konkrétně, je třeba zmínit i možnosti použití takzvaných bezpečnostních PLC, neboli safety PLC.

Safety PLC je řídicí jednotka schopná vykonávat stejné úkony jako běžný programovatelný automat. Ale zároveň splňuje přísné bezpečnostní nároky čímž, dosahuje na kvalifikaci kategorie SIL 3. SIL neboli Safety Integrity Level je mezinárodní bezpečnostní stupnice, která dle normy IEC 61508 udává 4 úrovně bezpečnosti, přičemž SIL 4 je úroveň s nejpřísnějšími požadavky. Pro dosažení bezpečnosti úrovně SIL 3 je přípustná chybovost zařízení zhruba 1 závažná chyba za 1000 let [21]. Aby mohlo být PLC certifikováno jako bezpečnostní musí projít kontrolu 3 strany zabývající se testováním a certifikací průmyslových zařízení. Pro Evropu se jedná o TÜV Rheinland [16] [17].

Bezpečnostní PLC má 4 části [16]:

- Modul vstupních signálů

Modul vstupních signálů je předřazen samotné řídicí jednotce, a co se týče bezpečnostních úkonů, je na ní nezávislý. V případě detekce selhání na vstupních signálech přejde do módu bezpečného vypnutí. Bezpečnost je mimo jiné zajištěna zpětnou diagnostikou vstupních signálů. Ta probíhá formou kontrolování frekvence pulsů vysílaných modulem vstupů do bezpečnostní sensoriky. Mezi další bezpečnostní prvky patří například zdvojené zapojení sensorů, kdy jsou pro každý bezpečnostní prvek přiděleny 2 vstupy.

- Bezpečnostní komunikační síť

Bezpečnostní komunikační síť (sběrnice) zajišťuje správnou komunikaci mezi jednotlivými periferiemi safety PLC a kontroluje, že přenášená informace dorazila do cíle, v nezměněném stavu a následně byla správně interpretována. Jako takováto bezpečnostní síť slouží Profisafe, což je v podstatě nadstavba komunikačních sítí Profibus určená pro bezpečnostní účely.

- Bezpečnostní řídicí jednotka

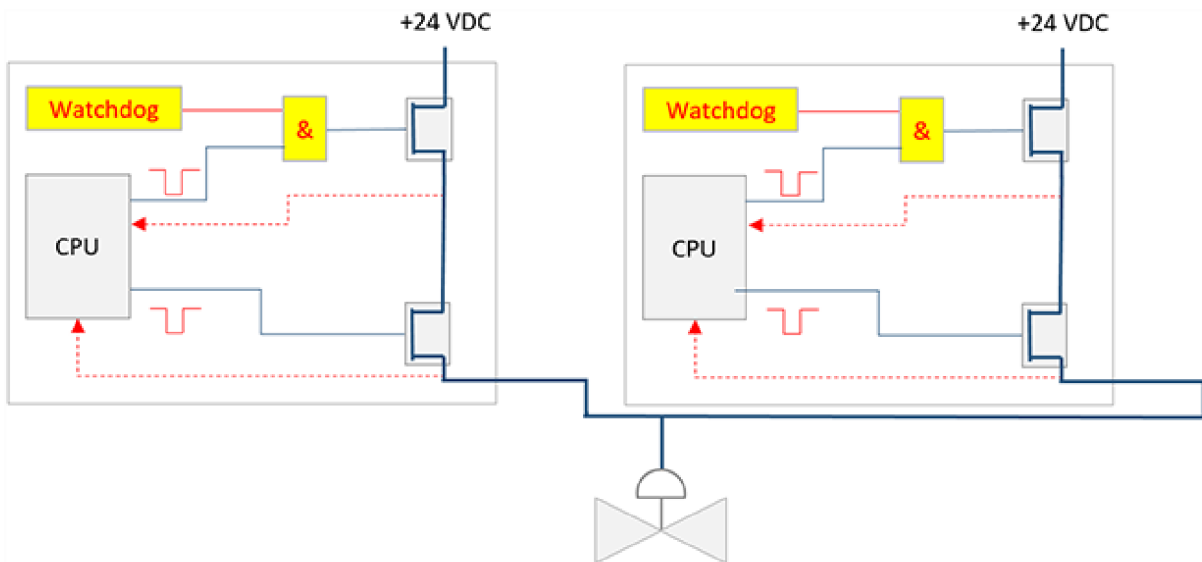
Je navržena tak, aby během samotného procesu řízení průběžně prováděla diagnostiku a v případě odhalení softwarové nebo hardwarové chyby okamžitě reagovala. Tato kontinuální kontrola je možná díky nadbytečným blokům, které jsou sice pro samotnou funkci řízení postradatelné, ale nezbytné pro získání požadované úrovně bezpečnosti.

- Modul výstupních signálů

V zásadě je jeho funkce velmi podobná modulu vstupních signálů. Rovněž je jednou z klíčových vlastností neustálá kontrola výstupních signálů pomocí kontrolních, periodicky se opakujících pulsů. Stejně tak při reakci na chybu nespolehá na řídicí jednotku, ale kroky směřující k zamezení kolize provede sám. Rovněž má zdvojené výstupy pro každý signál. V případě, že vyhodnotí na jednom výstupu chybu, je druhá větev použita k vypnutí připojené periferie. Zároveň má funkci

závěrečné kontroly celého řetězce. To znamená, že z výstupů jsou vyslány signály pouze v případě, že operace na všech třech předcházejících členech proběhly bezchybně.

Bezpečnostní PLC plní tedy stejnou funkci jako běžné PLC, s tím rozdílem že je minimální šance na selhání, a pokud už by k němu mělo tedy dojít, selže předvídatelným a bezpečným způsobem. Ve zkratce se dá hlavní příčinou takovéto úrovně bezpečnosti označit nadbytečnost. Neboli nadbytečný hardware, který není nezbytný pro funkci řízení, ale hraje nezbytnou roli v případě bezpečnosti. Na Obr. 3-5 lze vidět bezpečný výstupní obvod za použití dvou stejných výstupních modulů. Zároveň zde lze pozorovat znázornění pulsní kontroly výstupních signálů u každého modulu. Na rozdíl od běžného PLC není celá vnitřní logika volně přístupná a upravovatelná. Právě část týkající se bezpečnostní části programu je uzamčená a zapečetěná, aby nemohlo dojít k jejím neodborným a nebezpečným úpravám [17].



3-5) Paralelní zapojení dvou výstupních modulů. [22]

Zda použít či nepoužít bezpečnostní PLC ovlivňuje několik faktorů. Hlavním hlediskem je komplikovanost řízeného systému. Zatímco pro jednodušší systémy je vhodnější použít běžné PLC, pro systémy komplikovanější obsahující více bezpečnostních prvků už může být vhodné použít Safety PLC. To může přinést jak dlouhodobou finanční úsporu, tak snazší tvorbu a úpravu programu.

Výhody běžného PLC [18]:

- Jednoduché zapojení a řešení problémů
- Finančně výhodnější pro jednoduché systémy

Nevýhody běžného PLC [18]:

- Komplikovaná tvorba více e-stop zón

- Značně omezené možnosti sebediagnostiky
- Značně stoupající komplikovanost při složitějších systémech

Výhody bezpečnostního PLC [18]:

- Flexibilní
- Ideální pro více E-stop zón
- Dobré možnosti diagnostiky
- Bezpečnostní a běžné řízení v jedné jednotce
- Možná dlouhodobá úspora

Nevýhody bezpečnostního PLC [18]:

- Vyšší vstupní investice
- Nevhodné pro jednoduché systémy
- Potřebné další proškolení/rekvalifikace personálu

Přihlédneme-li ke konceptu výrobní buňky, jež je předmětem této práce a její relativní jednoduchosti – neobsahuje více e-stop zón a počet bezpečnostních prvků není vysoký, bude výsledný řídicí program koncipován při použití běžného PLC. O bezpečnost by se tedy staraly bezpečnostní prvky samotného stroje. V případě robotu by následně bylo možné použít například safety modul vstupů a výstupů [23], který by se staral o řízení zvolených bezpečnostních prvků. Například světelných závor nebo bezpečnostního zámku.

4 PRAKTICKÁ ČÁST

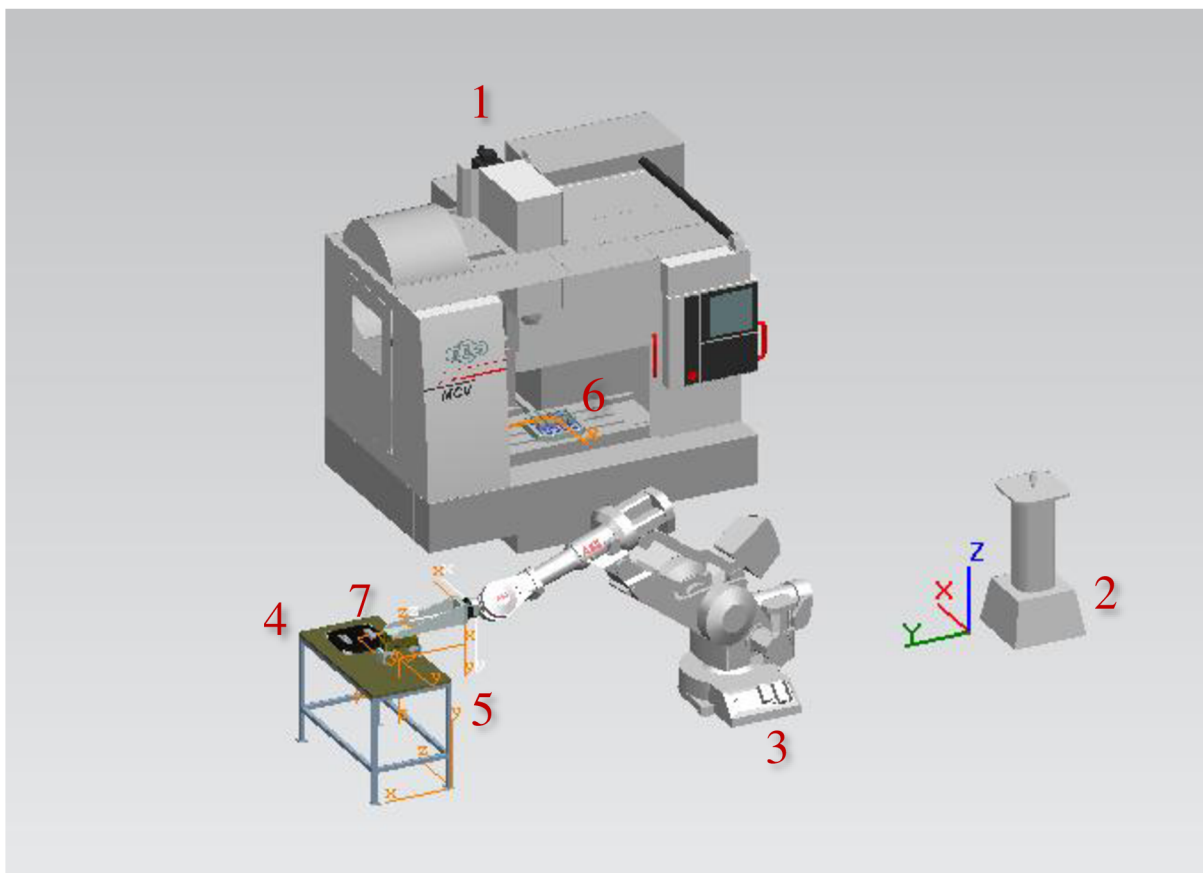
4.1 Zadané pracoviště

K zadané práci byl poskytnut jednoduchý model zpracovávané výrobní buňky sestavený v programu Process Simulate (dále jen PS), v pozdější fázi posloužil pro testování a ověření funkčnosti sestaveného kódu. Fakt, že došlo k použití tohoto modelu a následného propojení s PLC programem, do značné míry celou práci ovlivnil. O konkrétních případech bude pojednáno v nadcházejících kapitolách.

4.1.1 Model zadané výrobní buňky a pracovní cyklus

Model buňky (Obr. 4-1) sestává ze 4 základních prvků. Obráběcího centra – 1, stacionární brusky – 2, robotického manipulátoru – 3 a zakládací – 4 a odkládací stanice – 5, které byly v modelu zastoupeny formou stolu. Dále se v obráběcím centru nachází model magnetického upínače – 6, který je nezbytný pro umožnění zautomatizování celého výrobního procesu. Posledním prvkem, nacházejícím se v modelu je samotný obrobek – 7, který se nachází na zadní části stolu neboli zakládací stanici.

Model jako takový neobsahoval žádné bezpečnostní prvky, a proto byl pro sestavení programu jako informace o splnění/nesplnění bezpečnosti použit výstupní signál robotu. Viz kapitola 2.1.2 Zpracovávané signály



4-1) Model zadaného pracoviště v PS

Výrobní cyklus:




















Po počáteční inicializaci buňky, kdy dojde k uvedení do produkce schopného stavu všech jednotlivých přístrojů, může být spuštěn proces výroby. Ten je zahájen prvním úkonem robotu, kterým je uchopení obrobku na zakládací stanici a najetím před obráběcí centrum. Následně je obrobek robotem uložen na upínací desku a posléze manipulátor opustí pracovní prostor obráběcího centra. Nyní může proběhnout obrobení součásti. Jakmile je tato operace dokončena, je obrobek robotem opět vyňat a přemístěn mimo obráběcí centrum. Poslední operací, týkající se opracování obrobku, je odjehlení na stacionární brusce. Jakmile je i tento úkon úspěšně dokončen, dojde k odložení obrobku na odkládací stanici, čímž je zakončen jeden pracovní cyklus buňky. Během celého cyklu zároveň samozřejmě dochází k automatickému řízení zbylých prvků jako je včasné spuštění brusky, otevření a zavření dveří a zapnutí a vypnutí upínače tak, aby byly vždy splněny podmínky pro hladký průběh celého cyklu.

4.1.2 Zpracovávané signály

V modelu buňky byly již obsaženy některé signály a senzory. Jednalo se zejména o vstupy a výstupy týkající se řízení robotu a následně senzory informující o přítomnosti či případné nepřítomnosti obrobku v zakládací a odkládací stanici. Pro potřeby verifikace funkčnosti programu v pozdější fázi, kdy byl PLC program virtuálně připojen na model buňky, došlo k doplnění dalších signálů. Jednalo se o signály obráběcího centra a jeho dveří, magnetického upínače a stacionární brusky.

Signály robotu (Obr. 4-2):

V programu pro řízení robotu figuruje osm vstupních a jedenáct výstupních signálů. Krom základních signálů týkajících se inicializace robotu a jeho základních funkcí jako je spuštění motorů, nebo načtení programu se jednalo o pět výstupů volajících požadované operace – *OP_FromTable*, *OP_ToMachine*, *OP_FromMachine*, *OP_Grind*, *OP_PlaceOut*. Při jejich aktivaci dojde ke spuštění animace reprezentující danou úlohu manipulátoru.

 AutoSwitchOn	<input checked="" type="checkbox"/>				<input type="checkbox"/>
 irb4400_60_rev02_PN_I_AutoOn	<input checked="" type="checkbox"/>				<input type="checkbox"/>
 irb4400_60_rev02_PN_I_RobotWorking	<input checked="" type="checkbox"/>				<input type="checkbox"/>
 irb4400_60_rev02_PN_I_RoutineDone	<input checked="" type="checkbox"/>				<input type="checkbox"/>
 irb4400_60_rev02_PN_I_RunChainOK	<input checked="" type="checkbox"/>				<input type="checkbox"/>
 irb4400_60_rev02_PN_I_MotorsOnState	<input checked="" type="checkbox"/>				<input type="checkbox"/>
 irb4400_60_rev02_PN_I_MotorsOffState	<input checked="" type="checkbox"/>				<input type="checkbox"/>
 irb4400_60_rev02_PN_I_CycleOn	<input checked="" type="checkbox"/>				<input type="checkbox"/>
 irb4400_60_rev02_PN_Q_StartAtMain			<input checked="" type="checkbox"/>		<input type="checkbox"/>
 irb4400_60_rev02_PN_Q_MotorsOff			<input checked="" type="checkbox"/>		<input type="checkbox"/>
 irb4400_60_rev02_PN_Q_MotorsOn			<input checked="" type="checkbox"/>		<input type="checkbox"/>
 irb4400_60_rev02_PN_Q_Load			<input checked="" type="checkbox"/>		<input type="checkbox"/>
 irb4400_60_rev02_PN_Q_EmergencySt...			<input checked="" type="checkbox"/>		<input type="checkbox"/>
 irb4400_60_rev02_programPause			<input checked="" type="checkbox"/>		<input type="checkbox"/>
 OP_FromTable			<input checked="" type="checkbox"/>		<input type="checkbox"/>
 OP_ToMachine			<input checked="" type="checkbox"/>		<input type="checkbox"/>
 OP_FromMachine			<input checked="" type="checkbox"/>		<input type="checkbox"/>
 OP_Grind			<input checked="" type="checkbox"/>		<input type="checkbox"/>
 OP_PlaceOut			<input checked="" type="checkbox"/>		<input type="checkbox"/>

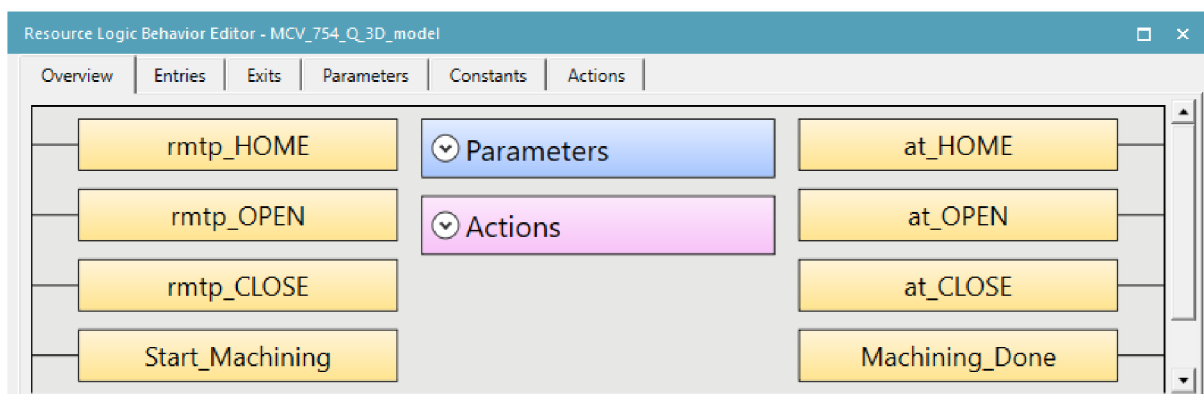
Vstupy

Výstupy

4-2) Signály poskytnuté s modelem

Signály obráběcího centra:

Signály v obráběcím centru modelu (Obr. 4-4), nebyly předem namapované a rovněž kromě ovládání dveří není v PS simulována jejich funkce. Proto pro potřeby ověření funkčního programu byly zhotoveny náhradní signály. Pro dveře byly využity signály z existujícího logického bloku (Obr. 4-3), takže možnost animace otevírání a zavírání dveří zůstala zachována. Rovněž byl použit výstupní signál pro spuštění obrábění. Vstupní signály *MC_OffState*, *MC_OnState*, *MC_ProgramEnded*, *MC_Working* jsou pak zastoupeny formou key signálů. Z toho důvodu během simulace je tedy nezbytná jejich manuální aktualizace. V případě kompletního modelu nebo reálné buňky, by aktualizace jejich stavu samozřejmě probíhala automaticky.



4-3) Logický blok pro obráběcí centrum v PS

	MCV_754_Q_3D_model_Start_Machining	
	MCV_754_Q_3D_model_rmtp_CLOSE	
	MCV_754_Q_3D_model_rmtp_OPEN	
	MCV_754_Q_3D_model_at_CLOSE	
	MCV_754_Q_3D_model_at_OPEN	
	MConState	
	MCOffState	
	Mc_Working	
	MC_ProgramEnded	

4-4) Signály obráběcího centra

Signály stacionární brusky:













Pro řízení stacionární brusky byla rovněž sestavena jednoduchá struktura signálů (Obr. 4-5). Jedná se o dva výstupní signály, *Grinder_Start* a *Grinder_Stop*, a dva vstupní signály, *Grinder_OnState* a *Grinder_OffState*, které jsou opět zastoupeny formou key signálů a dávají PLC informaci o stavu periferie.

	Grinder_GrinderStart		
	Grinder_GrinderStop		
	Grinder_ONState		
	Grinder_OffState		

4-5) Signály stacionární brusky

Zbylá sensorika a upínač:

Mezi poslední signály PS, které jsou použity pro sestavení řídicího kódu patří vstupy a výstupy magnetického upínače (Obr. 4-6), pro který byly signály sestaveny po vzoru stacionární brusky. Následně už zbývá pouze zmínit signály dvou senzorů (*InStation a OutStation*), snímajících přítomnost obrobku v zakládací a odkládací stanici.

 Schunk_SchUNKStart		
 Schunk_SchUNKStop		
 SchunkOnState		
 SchunkOffState		
 In.Sensor		
 Out.Sensor		

4-6) Signály upínače a senzory na zakládací a odkládací stanici

Signály Generate_Start a Kill_Start

Mezi signály se ještě objevují tyto dva. Jejich funkce je čistě simulační pro potřeby PS a v reálné buňce by nefigurovaly. Pomocí signálu *Generate_Start* se vygeneruje model obrobku na zakládací stanici a pomocí *Kill_Start* se následně odstraní ze stanice odkládací. Tyto signály jsou rovněž namapovány z PLC, takže je možná jejich aktualizace pomocí datového bloku. To by případně umožnilo přidání tlačítka na HMI panel, pomocí kterého by byly tyto operace řízeny. Jelikož se však navrhovaný kód měl co nejvíce blížit reálnému případu, nebyla tato možnost využita.

4.2 PLC program

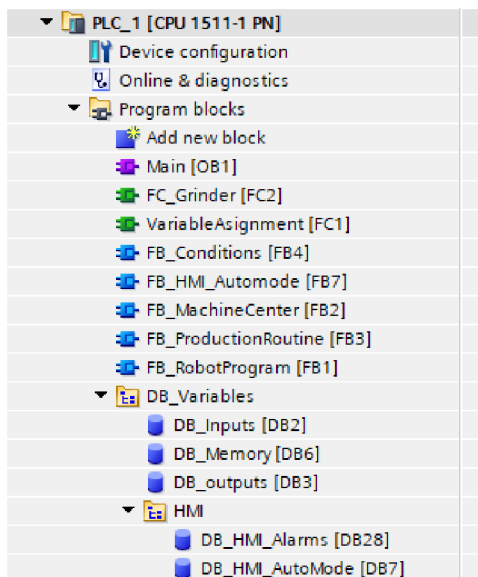
Řídicí kód byl sestaven v programu TiaPortal v15.1 firmy Siemens. Jako řídicí jednotka byla zvolena PLC Simatic S7-1511-1 PN. Hlavním důvodem volby byla možnost propojení s PS což by jinak pro podobnou úlohu dostačující jednotka z nižší řady 1200 neumožnila.

Celý kód byl čistě na základě vlastní preference sestaven výhradně pomocí strukturovaného textu neboli SCL. V následujících kapitolách je popsána a vysvětlena struktura programu, kterou doplňují vývojové diagramy, jež nabízí lepší přehlednost a názornost než program samotný. Vývojové diagramy jsou sestaveny tak, aby přehledně a srozumitelně popsaly hlavní části kódu a jejich architekturu. Z tohoto důvodu jsou záměrně vynechány některé prvky. Jedná se například o časovače, pomocné spínače atd. a některé redundantní podmínky a části kódu vložené pro zajištění bezchybného chodu programu.

4.2.1 Struktura řídicího kódu

Celý program vychází z jednotlivých funkčních bloků (Obr. 4-7), které jsou volány v hlavní rutině programu *MAIN*. Pro každou periférii, která má být řízena, byl sestaven vlastní funkční blok, jenž zajišťuje její řízení. Pouze bruska je řízena funkcí – *FC_Grinder*. V této části totiž nebylo potřeba definovat pomocné proměnné v blokovém rozhraní jako například v případě obráběcího centra. Další dva funkční bloky se starají o postupné volání operací jednotlivých periférií v případě, že se buňka nachází ve výrobním cyklu, a taky o ověřování podmínek, před voláním samotných operací. Tento způsob dává možnost snadno editovat a případně přenášet již sestavené kódy do dalších projektů. Rovněž je vhodný pro případ, kdy by následně bylo požadováno vzájemně nezávislé volání operací.

Ve fázi tvorby a následného odladění programu byly místo konkrétních tagů použity datové bloky, které vstupy a výstupy svou funkcí dočasně nahrazovaly. Před samotným propojením TiaPortalu s PS pak byla do hlavní rutiny vložena funkce, ve které došlo k přiřazení vstupů a výstupů k datovým blokům. Tento způsob umožnil snadné editování a průběžné testování programu bez nutnosti vytvářet tzv. Force Tables a jednalo se tak o uživatelsky příjemnější řešení. Další dva datové bloky se pak týkají HMI panelu. Celkově se tedy program skládá z pěti funkčních bloků a k nim přiřazených datových bloků, dvou funkcí a pěti uměle vytvořených datových bloků.



4-7) Funkční a datové bloky a funkce v TiaPortalu Datové bloky

DB_Memory

Vzhledem ke zvolené struktuře programu bylo potřeba definovat proměnné, na základě kterých mezi sebou jednotlivé funkční bloky komunikují. Hlavním takovým prvkem jsou proměnné z datového bloku DB_Memory (Obr. 4-8). Aktualizace těchto proměnných probíhá převážně ve funkčních blocích *FB_HMI_Automode* a *FB_ProductionRoutine*. Pro další funkční bloky pak slouží jako vstupy, na jejichž základě jsou pak nastaveny fyzické výstupy z PLC.

DB_Memory (snapshot created: 4/5/2021 5:43:35 PM)				
	Name	Data type	Start value	Retain
1	Static			<input type="checkbox"/>
2	b_OPCall_FromTable	Bool	false	<input type="checkbox"/>
3	b_OPCall_ToMachine	Bool	false	<input type="checkbox"/>
4	b_OPCall_MachiningS...	Bool	false	<input type="checkbox"/>
5	b_OpCall_FromMachi...	Bool	false	<input type="checkbox"/>
6	b_OPCall_Grind	Bool	false	<input type="checkbox"/>
7	b_OpCall_PlaceOut	Bool	false	<input type="checkbox"/>
8	b_ConditionsMet	Bool	false	<input type="checkbox"/>
9	b_OpCall_MCOpenDoor	Bool	false	<input type="checkbox"/>
10	b_OpCall_MCCloseDoor	Bool	false	<input type="checkbox"/>
11	b_OpCall_SchunkStart	Bool	false	<input type="checkbox"/>
12	b_OpCall_SchunkStop	Bool	false	<input type="checkbox"/>
13	b_OpCall_GrinderStart	Bool	false	<input type="checkbox"/>
14	b_OpCall_GrinderStop	Bool	false	<input type="checkbox"/>
15	b_Cell_CellStart	Bool	false	<input type="checkbox"/>
16	b_Cell_ProductionOn	Bool	false	<input type="checkbox"/>
17	b_Cell_StopProduction	Bool	false	<input type="checkbox"/>
18	b_Cell_StopCell	Bool	false	<input type="checkbox"/>
19	b_Cell_ProgramPause	Bool	false	<input type="checkbox"/>

4-8) Datový blok DB_Memory

DB Inputs a DB Outputs

Tyto datové bloky zastupují v programu fyzické vstupy a výstupy. Ve fázi přípravy a prvotního testování programu tak všude místo tagů figurovali proměnné těchto dvou souborů. Až ve chvíli, kdy docházelo k propojení TiaPortalu a PS byla vložena funkce FC_VariableAsi

4.2.2 FB_HMI_Automode

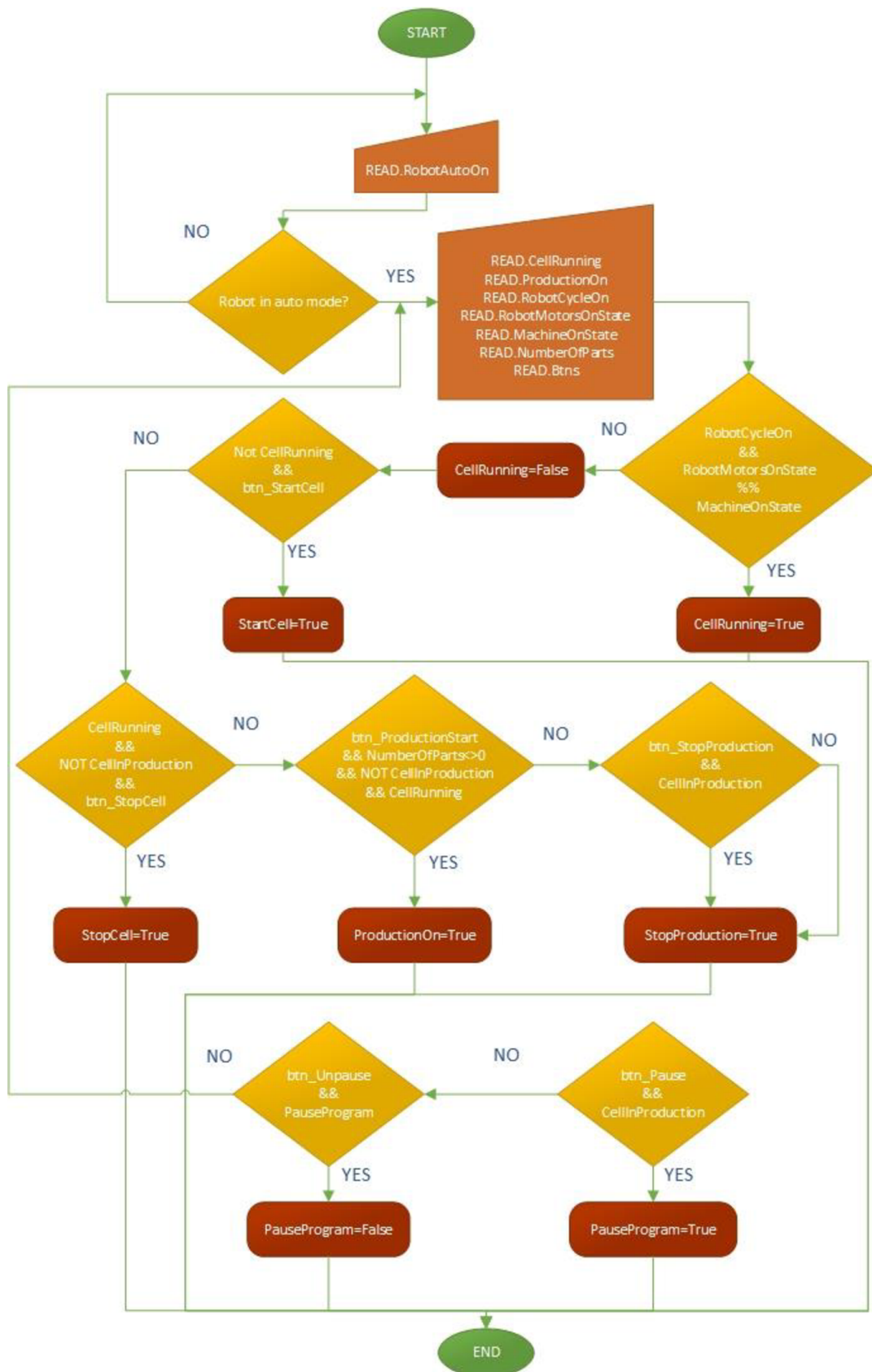
Blok FB_HMI_Automode je zásadní částí kódu, která zprostředkovává spojení mezi řídicím HMI panelem a částmi kódu, které se starají o samotnou produkci a volání operací jednotlivých periférií. Zároveň je tato část navržena tak, aby předcházela konfliktnímu volání operací při zmáčknutí tlačítka na displeji. Jako jeden z příkladů je možné uvést stav, kdy by se obsluha pokusila vypnout buňku uprostřed produkce. K tomuto stavu nesmí dojít a buňka lze vypnout pouze v případě, že se právě nenachází v režimu výrobní činnosti.

Aby vůbec bylo možné buňku řídit automatizovaně musí být manuálně přepnuta do automatického módu. V případě této práce se jedná o signál *RobotAutoOn*. Ve chvíli, kdy je tento signál aktivní, může dojít ke kontrole stavu vstupních proměnných, na základě kterých jsou poté nastaveny výstupy a volány operace. V této části programu se však jedná výhradně o proměnné uložené v datových blocích. Konkrétně pak v datovém bloku DB_Memory. Ty jsou pak dále zpracovávány v ostatních částech programu (jiných funkčních blocích), ve kterých teprve dochází k nastavení fyzických výstupů. Tímto kódem jsou tedy povelovány příkazy pro spuštění a vypnutí buňky, spuštění a vypnutí samotné výroby a pozastavení a opětovné spuštění produkce.

Součástí je rovněž logika starající se o aktualizaci stavu informačních diod (Obr. 4-9) na HMI panelu (viz kapitola 2.3.1 Vizualizace). Ve zbytku bloku jsou pak povelovány alarmy, informující uživatele o kolizních stavech při špatné volbě operací. Alarmů se týká kapitola 2.3.3.

```
86 REGION Kontrola diod pro MCdveže, Schunk a Brusku
87 // Statement section REGION
88
89 IF "DB_Inputs".b_G001_OnState THEN
90     "DB_HMI_AutoMode".b_HMI_D_Grinder := TRUE;
91 ELSIF "DB_Inputs".b_G001_OffState THEN
92     "DB_HMI_AutoMode".b_HMI_D_Grinder := FALSE;
93 END_IF;
94
95 IF "DB_Inputs".b_MC001_DoorOpened THEN
96     "DB_HMI_AutoMode".b_HMI_D_MCDoor := TRUE;
97 ELSIF "DB_Inputs".b_MC001_DoorClosed THEN
98     "DB_HMI_AutoMode".b_HMI_D_MCDoor := FALSE;
99 END_IF;
100
101 IF "DB_Inputs".b_Schunk001_OnState THEN
102     "DB_HMI_AutoMode".b_HMI_D_Schunk := TRUE;
103 ELSIF "DB_Inputs".b_Schunk001_OffState THEN
104     "DB_HMI_AutoMode".b_HMI_D_Schunk := FALSE;
105 END_IF;
106
107 IF "DB_Inputs".b_MC001_OnState THEN
108     "DB_HMI_AutoMode".b_HMI_D_MCState := TRUE;
109 ELSIF "DB_Inputs".b_MC001_OffState THEN
110     "DB_HMI_AutoMode".b_HMI_D_MCState := FALSE;
111 END_IF;
112 END_REGION
```

4-9) Aktualizace diod v FB_HMI_Automode



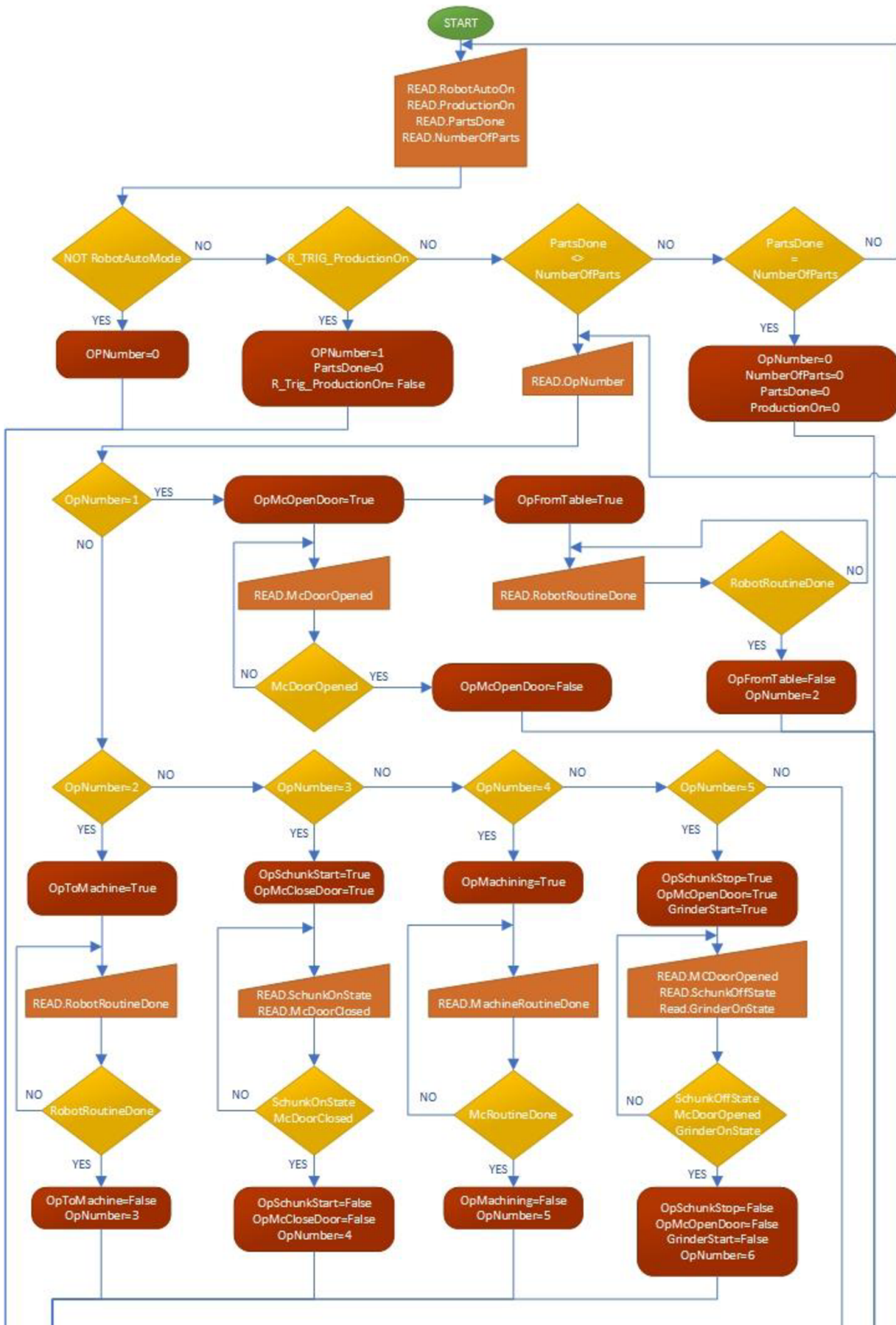
4-10) Vývojový diagram FB_HMI_Automode
(pozn. Signál *READ.Btns* reprezentuje načtení informací o všech tlačítkách na HMI panelu.)

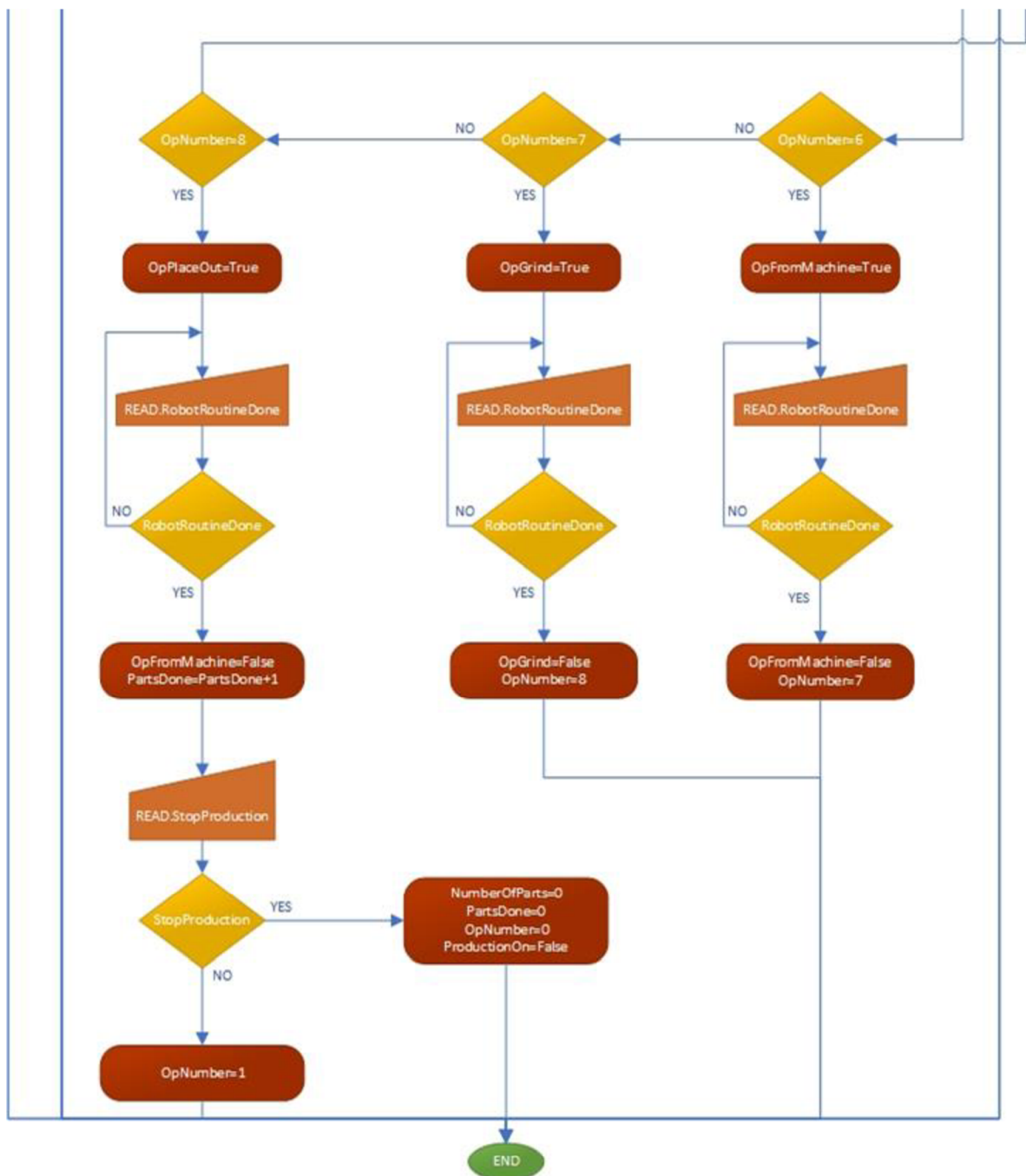
4.2.3 FB_ProductionRoutine

Tento funkční blok je celý vystaven kolem funkce *CASE*, která je spuštěna na základě pokynu k produkci, z *FB_HMI_Automode*. Přitom neustále sleduje proměnnou *OpNumber* a podle ní jsou volány operace, které má buňka vykonat. Hlavním úkolem této části programu je tedy zajistit cyklické volání operací v náležitém pořadí do doby, než je obsluhou výroba zastavena, nebo nebyl vyroben požadovaný počet součástí.

Po načtení hodnot proměnných dochází k ověření, že se robot nachází v automatickém režimu. Pokud tato podmínka není splněna, je signálu *OpNumber* přidělena hodnota 0. V tomto stavu jsou všechny operace nastaveny na *FALSE*. Jakmile přijde z bloku *FB_HMI_Automode* povel pro spuštění produkce, je proměnné *OpNumber* přiřazena hodnota 1 a zároveň vynulován počet hotových součástí *PartsDone*. Za předpokladu, že jsou splněny veškeré podmínky, je nyní volána první operace buňky. V počáteční fázi jsou volány operace pro otevření dveří obráběcího centra *OpMcOpenDoor* a první manipulační operace robotu, sebrání obrobku ze základací stanice – *OpFromTable*. V momentě, kdy jsou úspěšně dokončeny požadované operace dojde k resetu signálů *OpMcOpenDoor* a *OpFromTable* a přiřazení hodnoty *OpNumber=2*. Následně ve stejném formátu probíhá proces volání veškerých operací, během kterých dochází k průchodu obrobku celým výrobním procesem, obsaženém v této buňce.

V poslední fázi, kdy *OpNumber=8*, dochází k poslední operaci, a sice odložení obrobku na odkládací stanici. Během toho program ověří, že není aktivní požadavek pro zastavení produkce (Nedošlo ke stisknutí tlačítka *Stop Production*). V takovém případě by byly čítače požadovaných *NumberOfParts* a hotových částí *PartsDone* vynulovány, cyklus přerušen a buňka by čekala na zadání nové produkce. Pokud signál pro zastavení produkce není aktivní, dojde k přičtení jednoho hotového obrobku, tedy $PartsDone = PartsDone + 1$. V případě, že stále nebyl dosažen požadovaný počet součástí. Rozběhne se celý cyklus automaticky od začátku. Jakmile je zhotoven zadaný počet součástí, čítače se opět vynulují a buňka se opět nachází ve stavu před zahájením produkce.



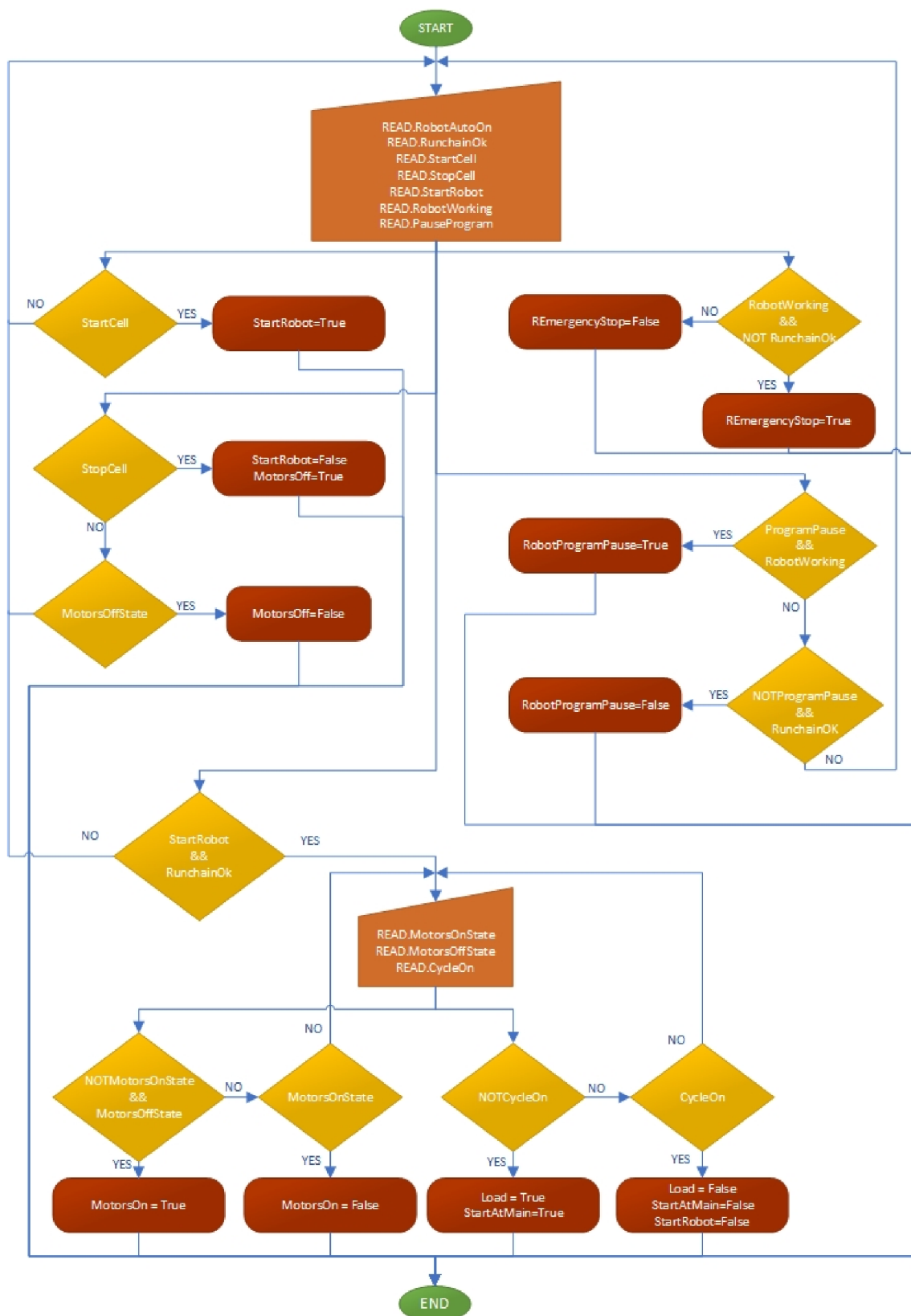


4-11) Vývojový diagram FB_ProductionRoutine

4.2.4 FB_RobotProgram

Řízení operací robotu je v této práci základem, na kterém stojí celý zbytek procesu. A právě k tomuto účelu slouží funkční blok *RobotProgram*. Pro zachování přehlednosti je popsán pomocí dvou vývojových diagramů. Popisují však části, které jsou v bloku psány na stejné úrovni.

První diagram (Obr. 4-12) znázorňuje příkazy týkající se inicializace a vypnutí robotu. Jedná se tedy o reakci na tlačítka pro vypnutí a zapnutí buňky, zapauzování a opětovné spuštění produkce a v neposlední řadě aktivace signálu *EmergencyStop* v případě porušení bezpečnosti. Jak již bylo zmíněno, předejití kolizních stavů a nevhodné volání operací je ošetřeno již ve funkčním bloku *FB_HMI_Automode*. Tato část programu již tedy pouze reaguje na volané funkce a nastavuje hodnoty výstupů, které by v reálu byly fyzicky připojeny k robotu.



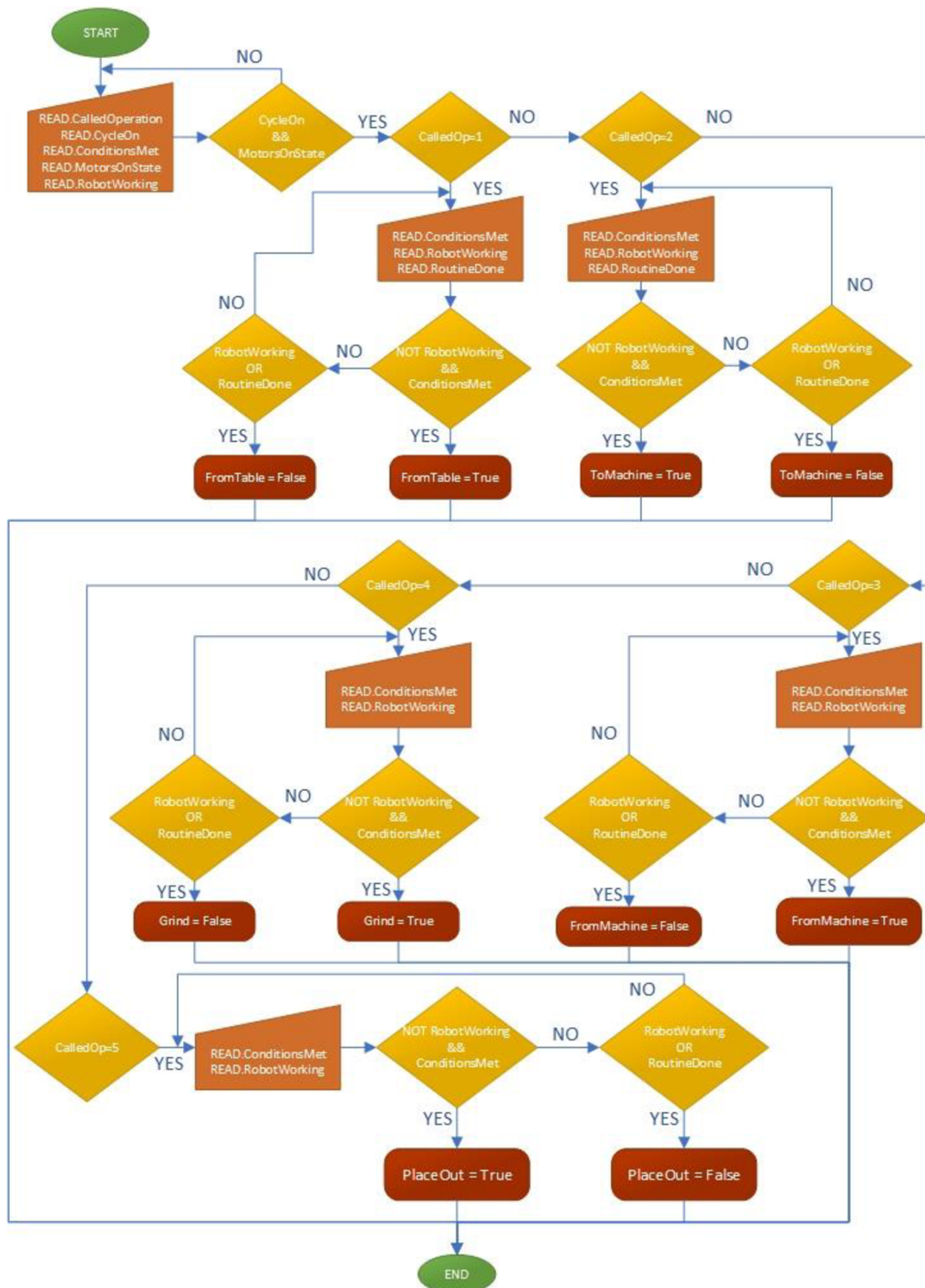
4-12) Vývojový diagram FB_RobotProgram – A

Druhý diagram (Obr. 4-14) představuje logiku, dle které se řídí manipulační procesy robotu. Pro tento účel bylo opět užito funkce CASE s proměnnou *CalledOperation*, jejíž

hodnota je nastavena na základě volané operace z bloku *FB_ProductionRoutine*. Přidělení čísel k jednotlivým operacím je vidět na obrázku. Každá operace běží podle stejného schéma. Nejprve je ověřeno, že robot právě nepracuje a jsou splněny podmínky pro danou operaci. O ověření podmínek se stará vlastní blok *FB_Conditions*. Pokud jsou tyto podmínky splněny je spuštěna operace robotu. Signál pro spuštění je deaktivován, jakmile PLC obdrží od robotu informaci, že pracuje neboli *RobotWorking=True* a zároveň je volání operace nulováno při aktivním signálu *RoutineDone* aby nedošlo k nechtěnému znovuspuštění této operace.

```
IF "DB_Memory".b_OPcall_FromTable THEN
    #nFB_CalledOperation := 1;
ELSIF "DB_Memory".b_OPcall_ToMachine THEN
    #nFB_CalledOperation := 2;
ELSIF "DB_Memory".b_OpCall_FromMachine THEN
    #nFB_CalledOperation := 3;
ELSIF "DB_Memory".b_OPcall_Grind THEN
    #nFB_CalledOperation := 4;
ELSIF "DB_Memory".b_OpCall_PlaceOut THEN
    #nFB_CalledOperation := 5;
ELSE
    #nFB_CalledOperation := 0;
END_IF;
```

4-13) Přiřazení hodnot pro jednotlivé operace



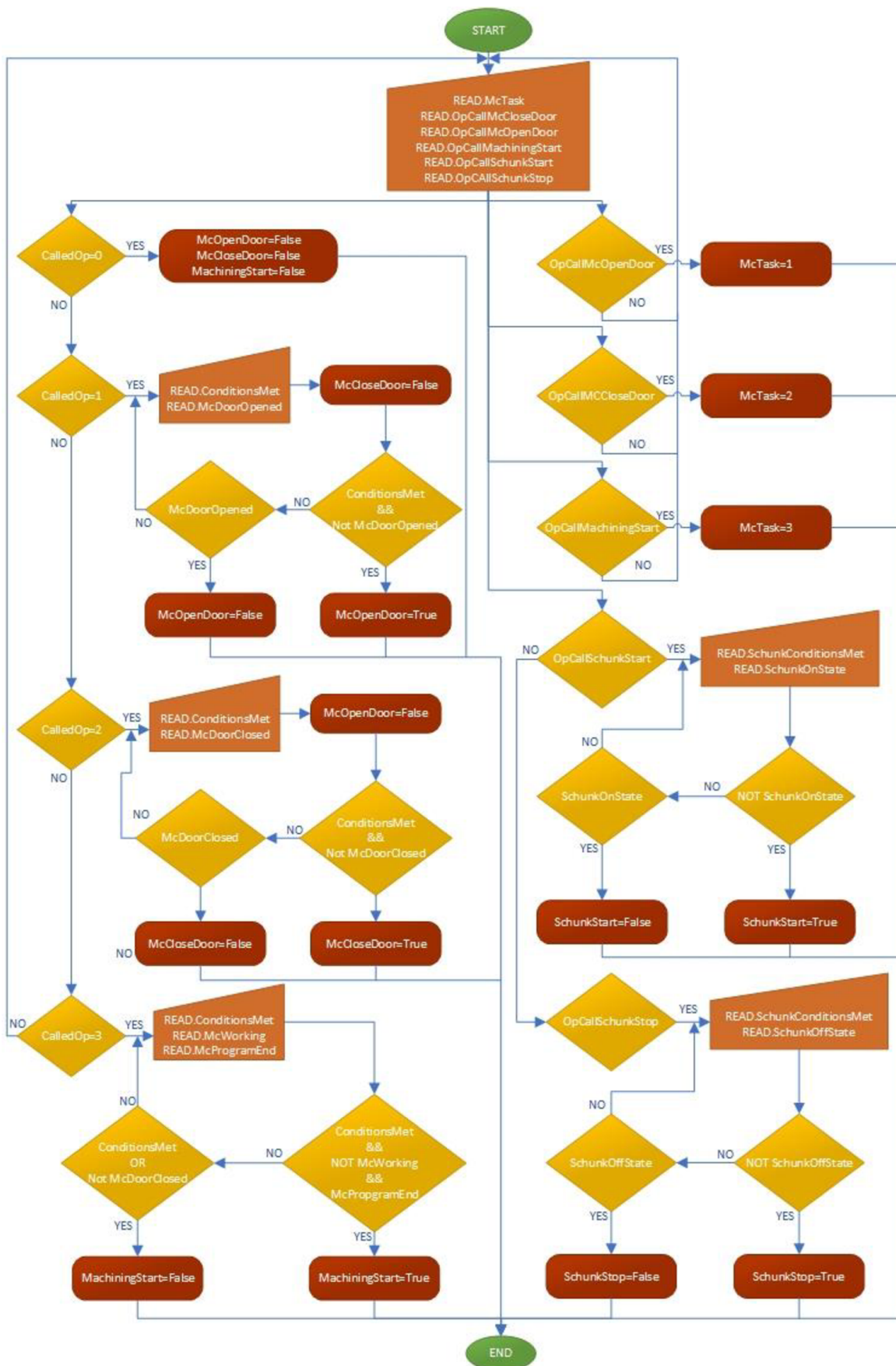
4-14) Vývojový diagram FB_RobotProgram – B

4.2.5 FB_MachineCenter

FB_MachineCenter je funkční blok starající se o řízení obráběcího centra. Vzhledem k poskytnutým signálům není součástí programu samotná inicializace a spuštění obráběcího centra. Je tedy předpokládáno, že tento proces již proběhl manuálně a OC pouze podává

informaci pro PLC, že se nachází v zapnutém stavu *MC_OnState*. Tento funkční blok tedy řídí spuštění obrábění a dále řízení dveří obráběcího centra a magnetického upínače, který se v něm nachází. Dle volané operace dochází k nastavení proměnné *McTask*, na základě které je následně ve funkci CASE volána požadovaná sekvence příkazů. Následně vše probíhá podle stejné logiky, kterou bylo možné sledovat v předešlých blocích.

Po načtení stavu vstupů dochází v případě splnění požadavků ke spuštění volané akce. Jakmile program zaznamená její úspěšné splnění je daný výstupní signál resetován. Z grafu je patrné, že ovládání upínače není zahrnuto ve funkci CASE. Jeden z důvodů je, že vzhledem k nenáročnosti logiky, podle které se ovládání upínače řídí není složité, bylo zbytečné zakládat další funkční blok, a tato část byla zahrnuta v kódu pro řízení OC. Hlavním důvodem je však zvolená struktura programu. Vrátime-li se k bloku *FB_ProductionRoutine*, dá se pozorovat stav, kdy dochází k současnému volání otevření dveří a vypnutí upínače a naopak. V případě, že by bylo ovládání upínače zahrnuto ve funkci CASE spolu s otevíráním a zavíráním dveří, vznikalo by tak konflikt, kdy by program měl snahu aktivovat 2 stavy CASEu, což samozřejmě není možné.



4-15) Vývojový diagram FB_MachineCenter

4.2.6 FB_Conditions

FB_Conditions je blok sloužící k ověřování podmínek před finální aktivací fyzických výstupů. Dochází-li tedy k inicializaci akce, pro kterou musí, nebo nesmí, být periferie buňky v konkrétním stavu, proběhne ověření právě zde. V současné podobě programu by se dal tento blok označit jako redundantní a své uplatnění by našel hlavně v případě, kdy by docházelo k manuálnímu volání operací, nikoliv k automatickému postupnému plynutí programu. Z povahy současné struktury kódu by totiž neměl stav, kdy dochází k volání operace bez splněných podmínek, vůbec nastat. To je zajištěno již v *FB_ProductionRoutine*, kdy přechod do další fáze (dalšího stavu funkce CASE) je podmíněn úspěšným zakončením všech úkolů stanovených ve fázi právě probíhající. Jedinou výjimku představují podmínky pro operace robotu, během kterých dochází k uchopení obrobku na zakládací stanici a na straně druhé pak k jeho odložení na konci cyklu – *OpCallFromTable* a *OpCallPlaceOut*. Ve chvíli, kdy bude prázdné zakládací nebo plné odkládací stanoviště, opravdu dochází ke kolizi a nesplnění podmínek. Zároveň díky faktu, že celá simulace v PS je bezchybová, jsou tyto dva stavy jediné případy, kde je možné demonstrovat chování buňky při nesplnění podmínek a zobrazení alarmu, který je přiřazený ke každé podmínce. Princip, dle kterého blok funguje je prostý a totožný s bloky předešlými.

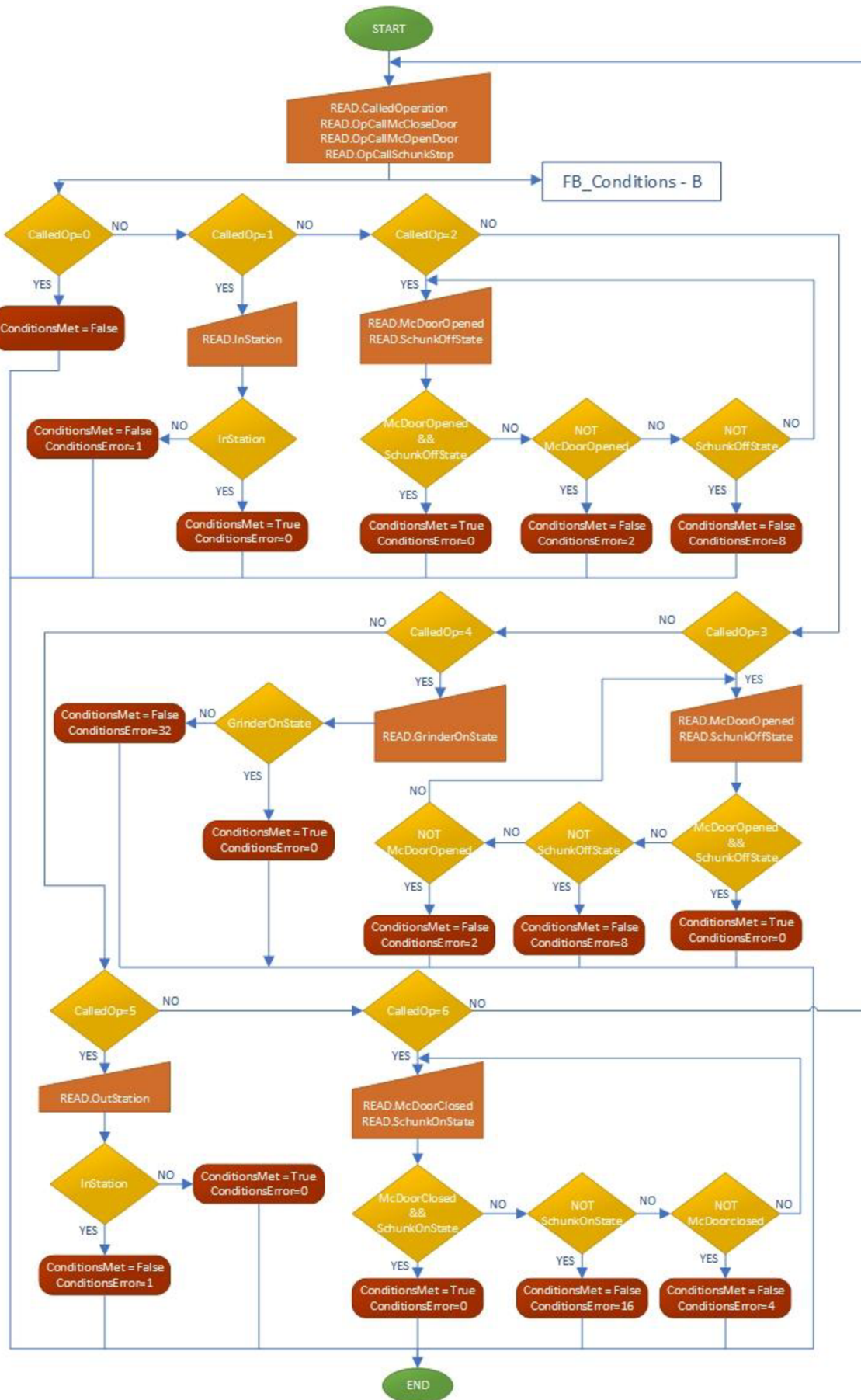
Dle volané operace je nastavena proměnná *CalledOp* pro CASE funkci (Obr. 4-16). Dle konkrétního stavu pak dochází k ověření požadovaných vstupních signálů a nastavení paměťové proměnné *ConditionsMet* na True nebo False dle vyhodnocení programu. V případě, že podmínky nejsou splněny je zároveň zobrazen alarm, informující obsluhu o podstatě vzniklé kolize.

```

1 IF "DB_Memory".b_OPcall_FromTable THEN
2   #nFB_CalledOperation := 1;
3 ELSIF "DB_Memory".b_OPcall_ToMachine THEN
4   #nFB_CalledOperation := 2;
5 ELSIF "DB_Memory".b_OpCall_FromMachine THEN
6   #nFB_CalledOperation := 3;
7 ELSIF "DB_Memory".b_OPcall_Grind THEN
8   #nFB_CalledOperation := 4;
9 ELSIF "DB_Memory".b_OpCall_PlaceOut THEN
10  #nFB_CalledOperation := 5;
11 ELSIF "DB_Memory".b_OPcall_MachiningStart THEN
12  #nFB_CalledOperation := 6;
13 ELSE
14  #nFB_CalledOperation := 0;
15 END_IF;

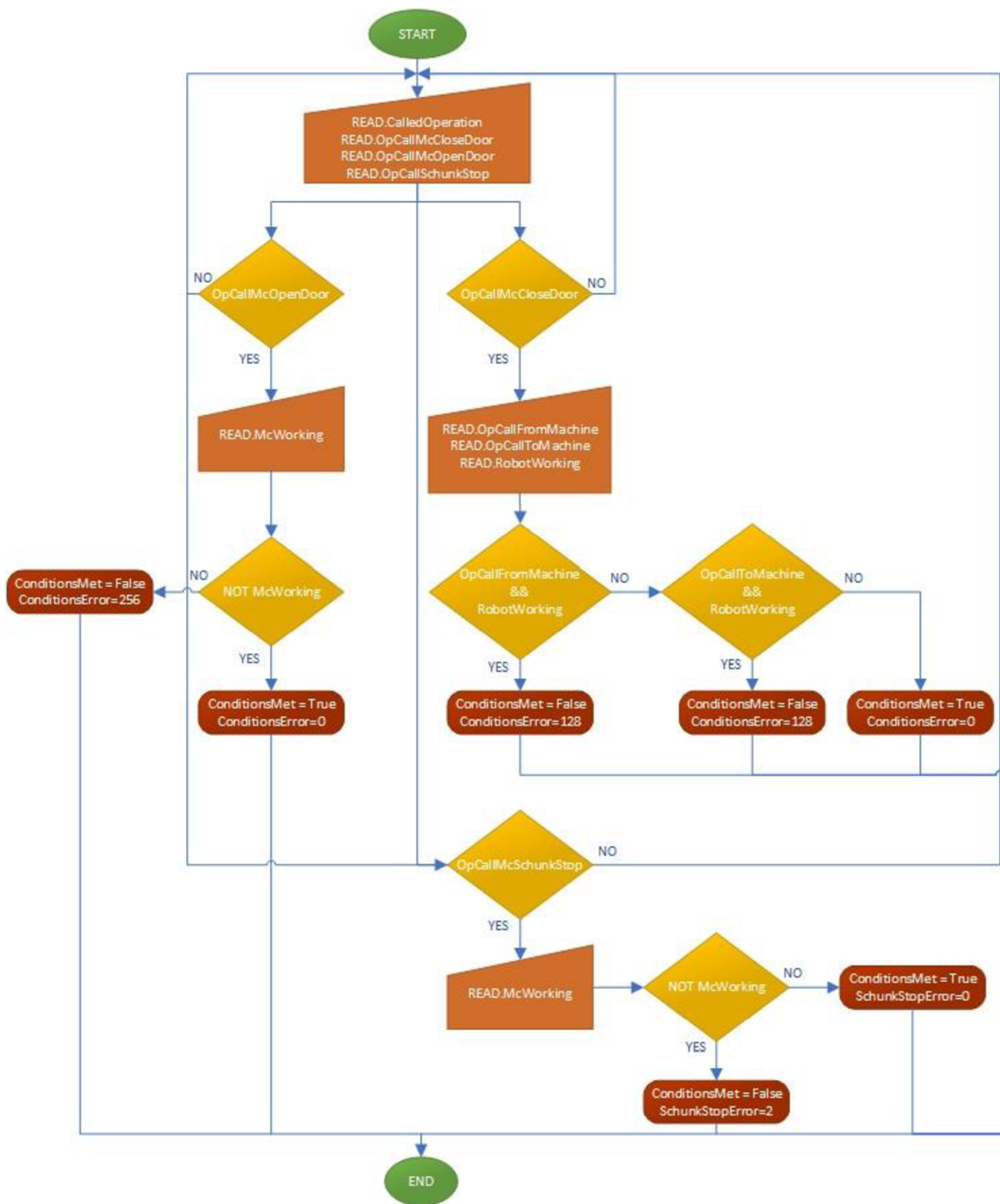
```

4-16) Přiřazení hodnot pro *CalledOperation* v *FB_Conditions*



4-17) Vývojový diagram FB_Conditions-A

Jak již bylo zmíněno v předešlé kapitole, v určité části programu dochází k současnému volání operací pro dveře obráběcího centra a magnetický upínač. Z toho důvodu je ověřování podmínek pro upínač řešeno samostatně, mimo funkci CASE a zároveň má přidělenou vlastní proměnnou pro zapsání stavu podmínek SchunkConditionsMet. Logika pro upínač je na stejné úrovni bloku jako funkce CASE. Pro zachování přehlednosti byl vytvořen vlastní vývojový diagram FB_Conditions – B (Obr. 4-18). Jeho umístění v FB_Conditions je zároveň znázorněno v diagramu FB_Conditions – A (Obr. 4-17).



4-18) Vývojový diagram FB_Conditions-B

4.2.7 FC_Grinder

FC_Grinder (Obr. 4-19) je nejjednodušší funkce zaslужující se o spouštění a vypnutí stacionární brusky. Strukturu bloku lze vidět na obrázku, kterou již vzhledem k jednoduchosti nepředstavuje vývojový diagram. Spustit brusku je možné pouze v případě, že je splněna bezpečnost, kterou v buňce zastupuje signál *RunchainOk*. Když je tato podmínka splněna čeká program na volání operace pro zapnutí brusky. V případě, že k této situaci dojde a bruska je vypnutá, dojde k aktivaci výstupu *G001_Start*, který je resetován ve chvíli, kdy *G001_OnState=True*.

Druhou situací, jež může nastat je volání operace pro vypnutí buňky. V tu chvíli je naopak aktivován výstupní signál *G001_Stop* a resetován v momentě, kdy PLC zaznamená, že *G001_OffState = True*. Navíc dojde k resetování proměnné pro volání operace *OpCall_GrinderStop*.

```
BP_WC_001_WithDB ▶ PLC_1 [CPU 1511-1 PN] ▶ Program blocks ▶ FC_Grinder [FC2]
IF... CASE... FOR... WHILE... (*...*) REGION R_TRIG IF... ELSE... F_TRIG
OF... TO DO... DO...
1 IF "DB_Inputs".b_RB001_RunchainOk THEN
2   IF "DB_Memory".b_OpCall_GrinderStart THEN
3     IF "DB_Inputs".b_G001_OffState THEN
4       "DB_outputs".b_G001_Start := TRUE;
5     ELSIF "DB_Inputs".b_G001_OnState THEN
6       "DB_outputs".b_G001_Start := FALSE;
7     END_IF;
8   ELSIF "DB_Memory".b_OpCall_GrinderStop THEN
9     IF "DB_Inputs".b_G001_OffState THEN
10      "DB_outputs".b_G001_Stop := FALSE;
11      "DB_Memory".b_OpCall_GrinderStop := FALSE;
12    ELSIF "DB_Inputs".b_G001_OnState THEN
13      "DB_outputs".b_G001_Stop := TRUE;
14    END_IF;
15
16  END_IF;
17
18 ELSE
19   "DB_outputs".b_G001_Stop := TRUE;
20 END_IF;
```

4-19) FB_Grinder

4.2.8 FC_VariablesAssignment

FC_VariablesAssignment (Obr. 4-20) je další jednoduchá funkce obsažená v kódu. Dochází v ní k přiřazení proměnných uložených v datových blocích *DB_Inputs* a *DB_Outputs* k PLC tagům. Tagy jsou rozděleny do čtyř samostatných souborů zvlášť pro Robot, obráběcí centrum, stacionární brusku a senzory na základací a odkládací stanici.

BP_WC_001_WithDB ▶ PLC_1 [CPU 1511-1 PN] ▶ Program blocks ▶ FC_VariableAssignment [FC1]

Block interface

IF...	CASE... OF...	FOR... TO DO...	WHILE... DO...	(*...*)	REGION	R_TRIG	IF... ELSE...	F_TRIG
-------	------------------	--------------------	-------------------	---------	--------	--------	------------------	--------

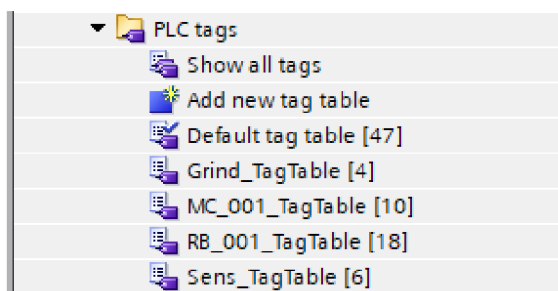
```

1  REGION Vstupy
2  //Robot
3  "DB_Inputs".b_RB001_AutoOn := "b_RB_001_I_AutoOn";
4  "DB_Inputs".b_RB001_RobotWorking := "b_RB_001_I_RobotWorking";
5  "DB_Inputs".b_RB001_RunchainOk := "b_RB_001_I_RunChainOK";
6  "DB_Inputs".b_RB001_MotorsOnState := "b_RB_001_I_MotorsOnState";
7  "DB_Inputs".b_RB001_MotorsOffState := "b_RB_001_I_MotorsOffState";
8  "DB_Inputs".b_RB001_CycleOn := "b_RB_001_I_CycleOn";
9  "DB_Inputs".b_RB001_RoutineDone := "b_RB_001_I_RoutineDone";
10
11 //Obráběcí centrum
12 "DB_Inputs".b_MC001_DoorOpened := "b_I_MC_001_DoorOpened";
13 "DB_Inputs".b_MC001_DoorClosed := "b_I_MC_001_DoorClosed";
14 "DB_Inputs".b_MC001_MCWorking := "b_I_MC_001_MCWorking";
15 "DB_Inputs".b_MC001_OnState := "b_I_MC001_OnState";
16 "DB_Inputs".b_MC001_OffState := "b_I_MC001_OffState";
17 "DB_Inputs".b_MC001_ProgramEnded := "b_I_MC_001_ProgramEnded";
18
19 //Upínací deska + senzory + bruska
20 "DB_Inputs".b_Schunk001_OnState := "b_I_Schunk001_OnState";
21 "DB_Inputs".b_Schunk001_OffState := "b_I_Schunk001_OffState";
22 "DB_Inputs".b_S001_InStation := "b_I_Sens001_InStation";
23 "DB_Inputs".b_S002_OutStation := "b_I_Sens002_OutStation";
24 "DB_Inputs".b_G001_OnState := "b_I_Grind_OnState";
25 "DB_Inputs".b_G001_OffState := "b_I_Grind_OffState";
26 END_REGION
27
28
29 REGION výstupy
30 // Robot
31 "b_RB_001_Q_FromTable" := "DB_outputs".b_RB001_FromTable;
32 "b_RB_001_Q_EmergencyStop" := "DB_outputs".b_RB001_EmergencyStop;
33 "b_RB_001_Q_FromMachine" := "DB_outputs".b_RB001_FromMachine;
34 "b_RB_001_Q_Grind" := "DB_outputs".b_RB001_Grind;
35 "b_RB_001_Q_PlaceOut" := "DB_outputs".b_RB001_PlaceOut;
36 "b_RB_001_Q_ProgramPause" := "DB_outputs".b_RB001_ProgramPause;
37 "b_RB_001_Q_StartAtMain" := "DB_outputs".b_RB001_StartAtMain;
38 "b_RB_001_Q_ToMachine" := "DB_outputs".b_RB001_ToMachine;
39 "b_RB_001_Q_MotorsOn" := "DB_outputs".b_RB001_MotorsOn;
40 "b_RB_001_Q_MotorsOff" := "DB_outputs".b_RB001_MotorsOff;
41 "b_RB_001_Q_Load" := "DB_outputs".b_RB001_Load;
42
43
  
```

4-20) Ukázka z FC_VariablesAssignment

4.2.9 PLC tagy

Tagy, či vstupní a výstupní fyzické proměnné jsou uloženy v tzv. Tag tables. Celkem došlo k vytvoření čtyř TagTable-ů jak je možné vidět na obrázku (Obr. 4-21).



4-21) Tag Tables

Názvy proměnných jsou sestaveny tak, aby jejich vyhledávání bylo logické a snadné. První písmeno představuje typ proměnné. Neboť jsou všechny vstupy i výstupy booleovské mají všechny předponu b. RB001 znamená zkrácené označení periferie, které se signál týká. V tomto případě RB001 jako robot číslo jedna. Následuje písmeno označující, zda se jedná o vstup – I nebo výstup Q. Název je zakončen krátkým textem charakterizujícím jeho funkci. Na obrázku (Obr. 4-22) je vidět TagTable obsahující signály pro robot.

BP_WC_001_WithDB ▶ PLC_1 [CPU 1511-1 PN] ▶ PLC tags ▶ RB_001_TagTable [18]								
RB_001_TagTable								
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervis...
1	b_RB001_I_AutoOn	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	b_RB001_I_RobotWorking	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	b_RB001_I_RoutineDone	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	b_RB001_I_RunChainOK	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	b_RB001_I_Motors OnState	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	b_RB001_I_Motors OffState	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	b_RB001_I_CycleOn	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	b_RB001_Q_FromTable	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	b_RB001_Q_ToMachine	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	b_RB001_Q_FromMachine	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	b_RB001_Q_Grind	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	b_RB001_Q_StartAtMain	Bool	%Q0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	b_RB001_Q_Motors On	Bool	%Q0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	b_RB001_Q_Motors Off	Bool	%Q0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	b_RB001_Q_Load	Bool	%Q1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	b_RB001_Q_EmergencyStop	Bool	%Q1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	b_RB001_Q_PlaceOut	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	b_RB001_Q_ProgramPause	Bool	%Q1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

4-22) RB001_TagTable

4.3 HMI panel

Další částí, která byla předpokladem úspěšného splnění zadání je návrh a vizualizace HMI panelu. Pro tyto účely byl použit displej TP1200 řady Comfort. Vzhledem k tomu, že součástí zadání nebyla bližší specifikace a požadavky na cenu, funkce atd., byl tento panel vhodnou volbou poskytující příjemné množství funkcí. Během práce byl kladen důraz, aby navržené řešení poskytovalo vše potřebné pro řízení zadané výrobní buňky a současně obsluze poskytovalo dostatek informací o jejím stavu. To vše při zachování přehlednosti a jednoduchosti.

S HMI panelem úzce souvisí dříve zmíněný funkční blok *FB_HMI_Automode* a datový blok *DB_HMI_Automode* (Obr. 4-23), který obsahuje proměnné spárované s tlačítky a informačními diodami panelu.

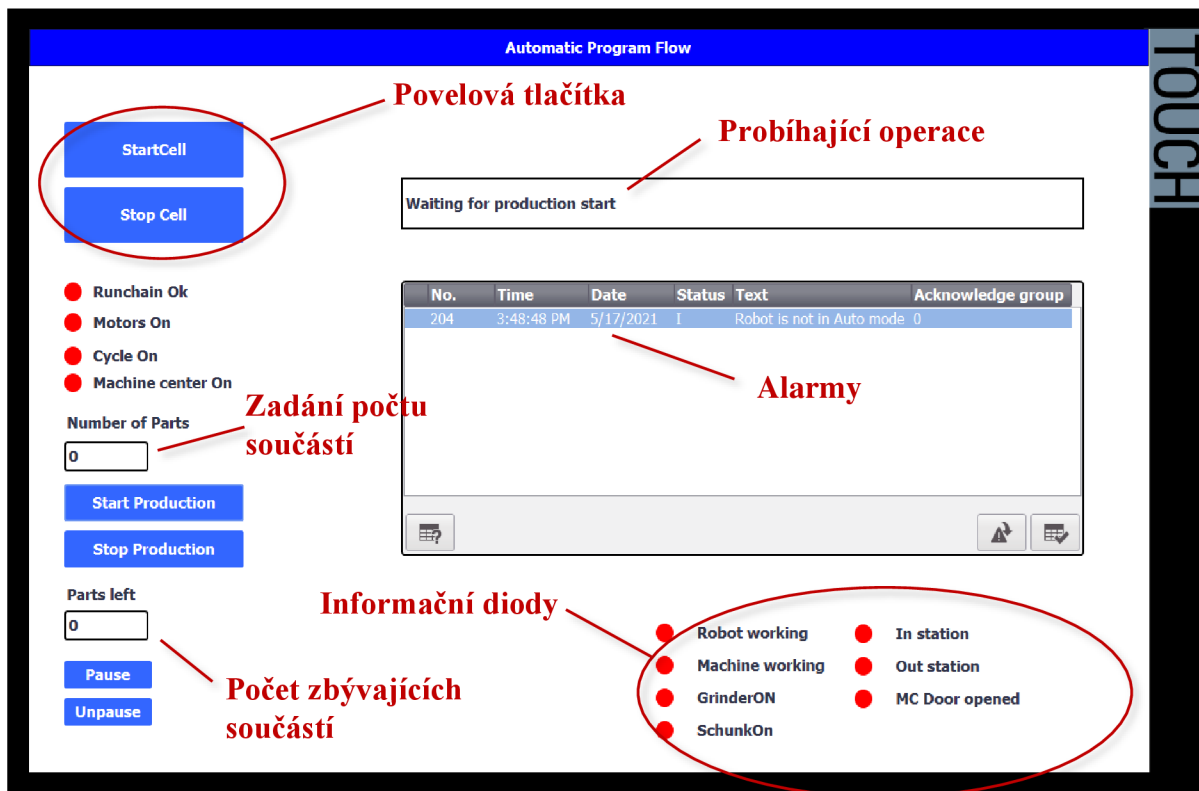
DB_HMI_AutoMode									
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	Static	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	b_HMI_btn_StartCell	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	b_HMI_btn_StopCell	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	b_HMI_btn_Productio...	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	b_HMI_btn_Pause at t...	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7	n_HMI_PartsLeft	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	n_HMI_NumberOfParts	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9	n_HMI_UndergoingOp	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	b_HMI_D_Grinder	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
11	b_HMI_D_Schunk	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
12	b_HMI_D_MCDoor	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
13	b_HMI_btn_Productio...	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
14	b_HMI_D_MCState	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
15	b_HMI_btn_PausePro...	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
16	b_HMI_btn_UnpauseP...	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

4-23) Datový blok DB_HMI_Automode

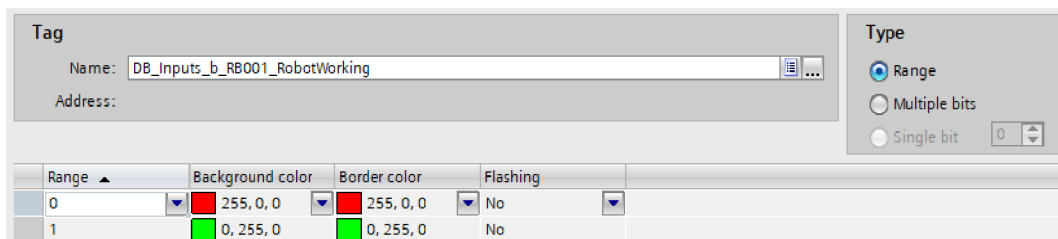
4.3.1 Vizualizace

Navržený panel (Obr.4-24) má jedinou obrazovku, která poskytuje dostatek prostoru pro umístění všech potřebných prvků. Z tohoto důvodu nebylo nutné navrhovat další Screensy.

K řízení buňky slouží šest tlačítek, přičemž se všechny nachází po levé straně panelu. Okénko s názvem *Number of Parts* slouží jako vstup, do kterého je zadán počet součástí, které mají být v následující sérii vyrobeny. Okénko *Parts left* pak operátora informuje o počtu součástí, zbývajících do konce právě probíhající produkce. Největší část obrazovky zabírají dvě informační pole. Horní, úzké pole dává ve formě textu informaci o právě probíhající operaci buňky. Spodní pole zobrazuje alarmy a výstrahy v případě chybových stavů programu. Alarmům a textovým informacím, které se mohou vyskytnout je věnována kapitola 2.3.3 *Alarmy a texty*. Posledním prvkem, který se na panelu nachází jsou informační diody. (Na obrázku červené puntíky). V případě, že s nimi spárovaný signál má hodnot *TRUE* změni svou barvu na zelenou.



4-24) Vizualizace HMI panelu



4-25) Ukázka definování vzhledu informační diody

4.3.2 Alarmy a texty

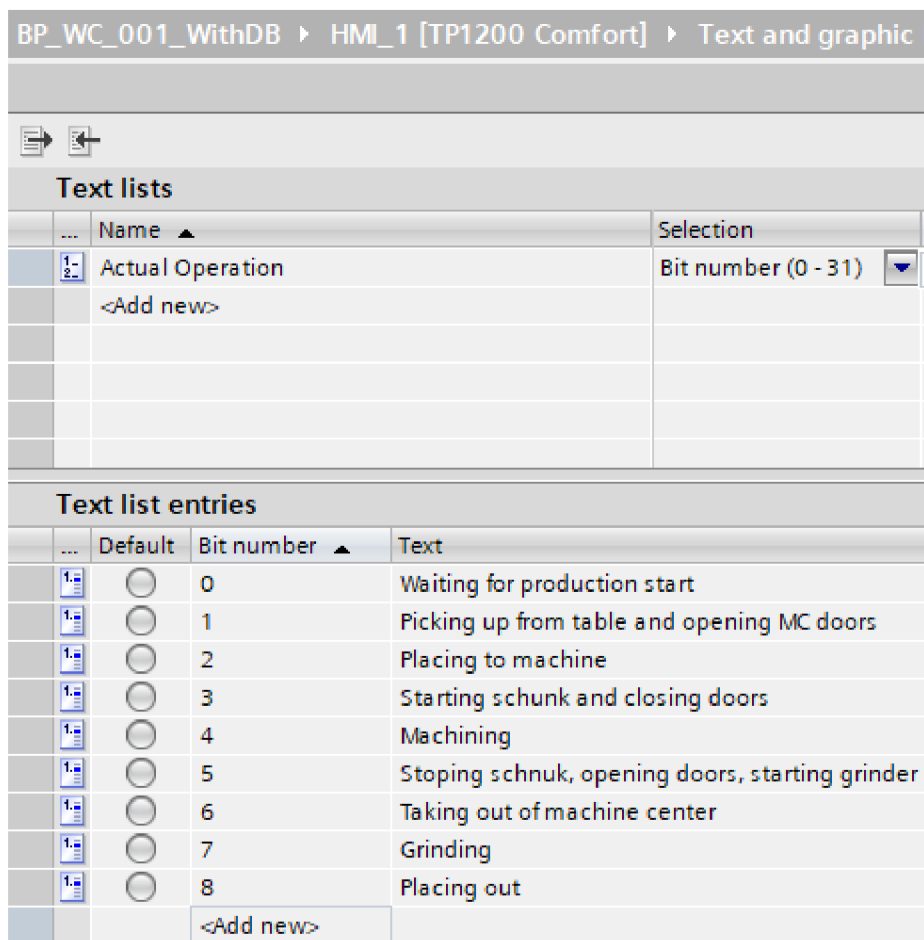
K informaci o právě probíhajícím úkonu během produkce slouží úzké textové pole na obrazovce řídicího panelu. V případě, že se zrovna buňka nenachází ve fázi produkce, je vyobrazen text *Waiting for production start*. Jakmile se však rozběhne výrobní cyklus, je vždy zobrazen text dle právě probíhající operace (Obr.4-26). Stav textu je řízen nastavením proměnné datového typu Integer, v rámci bloku FB_ProductionRoutine. Všechny varianty zobrazovaného textu znázorňuje obrázek 4-27.

```

6: //Vyjmtí z OC
   "DB_HMI_AutoMode".n_HMI_UndergoingOp := 64;
   "DB_Memory".b_OpCall_FromMachine := TRUE;

   IF "F_TRIG_R001_RoutineDone".Q THEN
     "DB_Memory".b_OpCall_FromMachine := False;
     #nFB_OPNumber := 7;
   END_IF;
  
```

4-26) Ukázka aktivace textu z funkce FB_ProductionRoutine



4-27) Text list pro první informační okno

Druhé textové okno slouží k zobrazení alarmů. V této práci bylo užito dvou typů alarmů. Errorů a tzv. warnings. Zatímco alarmy typu warnings samy zmizí ve chvíli, kdy stav podmiňující daný alarm skončí, error zůstává v poli alarmů, dokud obsluha nepotvrdí, že si je nastalého stavu vědoma. Errory tedy představují hlášení o chybových stavech, zatímco druhý typ alarmu slouží pouze jako varování, či informace, například při špatné volbě tlačítka na panelu. Seznam alarmů (Obr.4-28) Doplnjuje tabulka 1, kde jsou popsány podmínky pro aktivaci daného alarmu. Alarmy 101-109 jsou aktivní v případě, že dochází k volání některé z operací buňky, ale podmínky pro ni nejsou splněny. Vždy tudíž nastává pouze jeden z těchto

případů. Proto jsou alarmy namapovány na jednu proměnnou o různých hodnotách. Zbylým alarmům pak byla pro lepší přehlednost při psaní kódu a přípustnosti aktivace dvou a více alarmů zároveň, přiřazena vlastní proměnná. Veškeré proměnné pro alarmy jsou uloženy zvlášť v datovém bloku *DB_HMI_Alarms*. Vzhledem k tomu, že simulace z PS je bezchybová k aktivaci errorů v rámci této práce nedojde. Přesto jsou součástí kódu tak, jako by se jednalo o reálnou buňku.

BP_WC_001_WithDB ▶ HMI_1 [TP1200 Comfort] ▶ HMI alarms

Discrete alarms Analog alarms Controller alarms

Discrete alarms

ID	Name	Alarm text	Alarm class	Trigger tag	Trigge..	Trigger address
101	Empty In-station	In station is empty	Errors	DB_HMI_Alar...	0	DB_HMI_Alar...
102	Mc doors not open...	Mc doors are not opened	Errors	DB_HMI_Alar...	1	DB_HMI_Alar...
103	Mc doors not closed	Mc doors are not closed	Errors	DB_HMI_Alar...	2	DB_HMI_Alar...
104	Schunk not off	Schunk is not off state	Errors	DB_HMI_Alar...	3	DB_HMI_Alar...
105	Schunk not on	Schunk is not on state	Errors	DB_HMI_Alar...	4	DB_HMI_Alar...
106	Grinder not on	Grinder is not On state	Errors	DB_HMI_Alar...	5	DB_HMI_Alar...
107	Out-Station not em...	Empty station is not empty	Errors	DB_HMI_Alar...	6	DB_HMI_Alar...
108	Robot still working	Robot is still working	Errors	DB_HMI_Alar...	7	DB_HMI_Alar...
109	Machine Center still..	Machine center is still working.	Errors	DB_HMI_Alar...	8	DB_HMI_Alar...
201	Runchain not OK	Runchain is not OK	Errors	DB_HMI_Alar...	1	DB_HMI_Alar...
202	Mottors not on	Mottors start failed	Errors	DB_HMI_Alar...	1	DB_HMI_Alar...
203	Cycle not on	Robot is not in cycle	Errors	DB_HMI_Alar...	1	DB_HMI_Alar...
301	Cell in production ...	Cannot stop cell while production is	Warnings	DB_HMI_Alar...	1	DB_HMI_Alar...
302	Cell is running	Cell is already running	Warnings	DB_HMI_Alar...	1	DB_HMI_Alar...
303	Number of parts	Please assign number of parts to be	Warnings	DB_HMI_Alar...	1	DB_HMI_Alar...
304	Stopping Production	Production will be stopped at the en	Warnings	DB_HMI_Alar...	1	DB_HMI_Alar...
204	Robot Auto mode e...	Robot is not in Auto mode	Warnings	DB_HMI_Alar...	1	DB_HMI_Alar...
305	Cell stopped	Cell is not running	Warnings	DB_HMI_Alar...	1	DB_HMI_Alar...
306	Cell working	Cell is already working	Warnings	DB_HMI_Alar...	1	DB_HMI_Alar...
307	CellNotInProduction	Cell is not in production	Warnings	DB_HMI_Alar...	1	DB_HMI_Alar...
308	CellNotPaused	Cell is not paused	Warnings	DB_HMI_Alar...	1	DB_HMI_Alar...

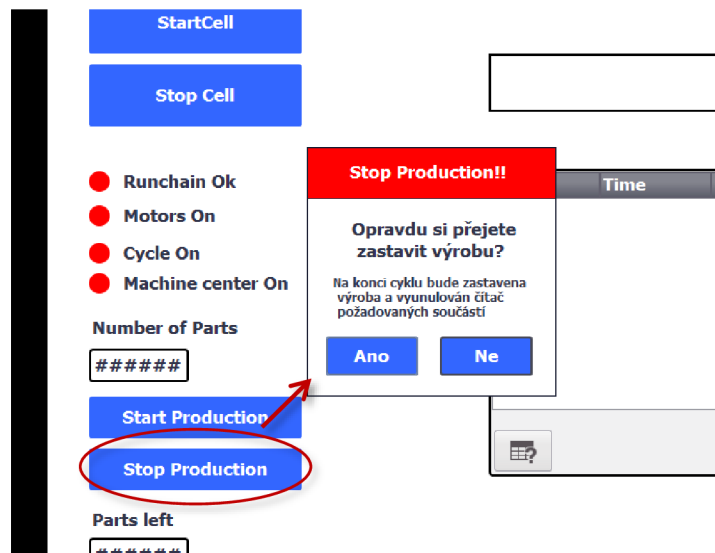
4-28) HMI Alarmy panelu

101	Empty In-station	Je volána operace FromTable, ale v základací stanici není obrobek
102	Mc doors not opened	Je volána operace, která vyžaduje otevřené dveře obráběcího centra, ale není tomu tak.
103	Mc doors not closed	Je volána operace, která vyžaduje zavřené dveře obráběcího centra, ale není tomu tak.
104	Schunk not off	Je volána operace, která vyžaduje vypnutý upínač, ale není tomu tak.
105	Schunk not on	Je volána operace, která vyžaduje zapnutý upínač, ale není tomu tak.
106	Grinder not on	Je volána operace Grind, ale bruska není zapnutá
107	Out-Station not empty	Je volána operace PlaceOut, ale není volná odkládací stanice
108	Robot still working	Je volána operace, která vyžaduje, aby robot nepracoval, ale není tomu tak.
109	Machine Center still working	Je volána operace, která vyžaduje, aby obráběcí centrum nepracovalo, ale není tomu tak.
201	Runchain not OK	Po spuštění buňky není splněna podmínka RunchainOK=True
202	Mottors not on	Po spuštění buňky není nedošlo ke spuštění motorů.
203	Cycle not on	Po spuštění buňky se robot nedostal do stavu CycleOn=True
204	Robot Auto mode error	Robot není přepnut do automatického režimu
301	Cell in production mode	Bylo stisknuto tlačítko pro vypnutí buňky během probíhající produkce.
302	Cell is running	Bylo stisknuto tlačítko pro zapnutí buňky, která je již zapnutá.
303	Number of parts	Obsluha se pokusila spustit výrobu s 0 požadovanými součástmi.
304	Stopping Production	Bylo stisknuto tlačítko pro zastavení probíhající produkce.
305	Cell stopped	Bylo stisknuto tlačítko pro vypnutí buňky, která není zapnutá.
306	Cell in production	Bylo stisknuto tlačítko pro spuštění produkce, ale ta již běží.
307	CellNotInProduction	Bylo stisknuto tlačítko pro zastavení produkce, ale ta neběží.

Tabulka 1) Seznam alarmů a podmínek jejich aktivace

4.3.3 Pop-up screens

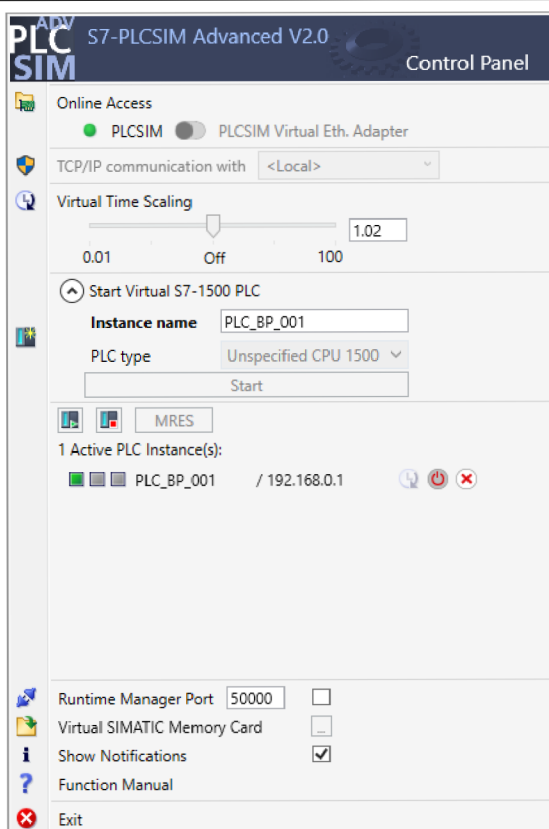
Aby nedocházelo k nechtěnému zadávání příkazů při neopatrném zmáčknutí tlačítka, byla použita funkce tzv. Pop-up screens (Obr.4-29) neboli vyskakovacích oken, kterou umožňuje použití panelu z řady comfort. V případě stisknutí tlačítka, se zobrazí vyskakovací okno, kde je obsluha dotázána, zda si přeje danou operaci opravdu provést a je vyzvána k jejímu potvrzení. V případě, že zmáčknutí nebylo úmyslné, má operátor možnost bez jakékoliv další akce vyskakovací okno zavřít.



4-29) Ukázka pop-up screenu pro Stop production

4.4 Virtuální zprovoznění

V pozdějších fázích testování programu bylo potřeba propojit sestavený program s modelem poskytnutým v Proces Simulate. Jako prostředník sloužil další program z portfolia značky Siemens a to S7 PLSSIM Advanced v2.0, pomocí kterého byla vytvořena virtuální PLC instance (Obr.4-30) na kterou byly následně kód sestavený v TiaPortalu a model v PS připojeny. V tuto chvíli bylo možné program testovat a odhalit a následně odstranit případné chyby.



4-30) Vytvořená PLC instance v PLCSIM Advanced

Posledním úkonem nutným k úspěšnému virtuálnímu zprovoznění tak bylo správné namapování signálů v PS na vstupy a výstupy TiaPortalu. Na obrázku (Obr.4-31) lze vidět názornou ukázkou. Po připojení PS na virtuální instanci tak mohla být spuštěna simulace a program otestován.

Signal Name	Memory	Type	Robot Signal Name	Address	IEC Format	PLC Connection	External Connection	Resource
irb4400_60_rev02_PN_I_RobotWorking	<input type="checkbox"/>	BOOL	I_RobotWorking	0.1	I0.1	<input checked="" type="checkbox"/>	PLC_BP_001	irb4400_60_rev02
irb4400_60_rev02_PN_I_RoutineDone	<input type="checkbox"/>	BOOL	I_RoutineDone	0.2	I0.2	<input checked="" type="checkbox"/>	PLC_BP_001	irb4400_60_rev02
OP_FromTable	<input type="checkbox"/>	BOOL	Q_OP_FromTable	0.0	Q0.0	<input checked="" type="checkbox"/>	PLC_BP_001	irb4400_60_rev02
OP_ToMachine	<input type="checkbox"/>	BOOL	Q_OP_ToMachine	0.1	Q0.1	<input checked="" type="checkbox"/>	PLC_BP_001	irb4400_60_rev02

4-31) Namapované signály PS

5 ZHODNOCENÍ A DISKUZE

Tato práce se zabývá sestavením kódu a řídicího HMI panelu pro zadané pracoviště. Dá se prohlásit, že navržený program splňuje požadavky na řízení buňky obsažené v zadání. Je zde však prostor pro případné rozšíření o zpracování dalších prvků, kterými by mohly být zejména prvky bezpečnosti. Některé části programu by dle požadavků případného koncového uživatele šlo zpracovat jiným způsobem. Například v případě stacionární brusky by bylo možné pouze řídit zrychlení a zpomalení jejích otáček před a po operaci broušení. Posouzení výhod a nevýhod tohoto řešení však není předmětem problematiky návrhu programu, nýbrž posouzení ekonomické a finanční bilance.

Navržená struktura programu umožňuje editaci, případně rozšíření o další dílčí kroky. Rovněž je vhodná pro případ, že by součástí řešení byl další screen HMI panelu, kde by docházelo k volání operací manuálně a vzájemně nezávisle, nikoliv v postupném navazujícím cyklu. Při tvorbě programu byla tato varianta uvažována, avšak po zhodnocení přínosu a časové dotace, nebyla nakonec realizována. Mezi další výhody patří lepší optimalizace cyklu, kdy nedochází ke zbytečným prodlevám, ale některé úkony jako například otevření dveří a spuštění stacionární brusky jsou realizovány simultánně. Na druhou stranu je z tohoto důvodu nezbytné použití více proměnných viz. separátní proměnná pro splnění podmínek při řízení upínače.

Zmíněná optimalizace by případně mohla být posunuta na vyšší úroveň za podmínek, že by byly známy některé nespecifikované údaje, jako například potřebný čas pro zapnutí brusky. Vhodným nastavením časového spínání by tak bylo možné cyklus ještě urychlit.

Jak již bylo zmíněno. Použité PLC z řady 1500 je pro tuto úlohu zbytečně výkonné a více než dostatečně by posloužilo PLC nižší řady což by přineslo finanční úsporu. Nicméně uvážíme-li poměr ceny PLC a zbylých periférií buňky jednalo by se zanedbatelnou částku. Je otázkou, zda ušetřené peníze vyváží absenci funkcí a možností, které může PLC vyšší řady poskytnout v budoucnosti.

V praxi by pak bylo vhodné začít program tvořit až na základě předem připravené a kompletní struktury signálů, odvíjející se od konkrétní volby hardwaru. Je tak zřejmé, že návrh řídicího programu je sice zásadním, ale malým článkem velkého celku.

6 ZÁVĚR

V rámci této bakalářské práce byl spolu s HMI panelem navržen řídicí PLC kód pro zadanou výrobní buňku. Poskytnutý model byl zpracován v programu Process Simulate firmy Siemens. Ke tvorbě programu a panelu byl tedy použit program TiaPortal téhož výrobce. K následnému propojení došlo za pomoci PLCSIM Advanced.

V teoretické části došlo k zasazení zpracovávané problematiky do celku virtuálního provozu. Následně představené kapitoly obecné údaje a pojmy týkající se hardwaru a softwaru PLC a HMI panelů. Postupně byly představeny některé z konkurenčních firem, typy a možnosti hardwaru i možnosti softwaru. Čtenář je seznámen se základními prvky, které mohou v podobném pracovišti figurovat a jakým způsobem celý proces ovlivní.

Počátkem praktické části dochází k seznámení se zadáním, jakožto základu, od kterého se následná tvorba programu odvíjela. Následuje vysvětlení struktury, která byla dodržena při psaní kódu a jednotlivé bloky jsou popsány v samostatných kapitolách. Ty jsou doplněny o názorné ukázky částí kódu a vývojové diagramy, ve kterých je možné přehledně a logicky sledovat tok programu.

Další kapitola je věnována návrhu HMI panelu. V podkapitolách jsou prostřednictvím obrázků a tabulky shrnuty a vysvětleny veškeré funkce v návrhu obsažené.

Závěrem lze konstatovat, že práce zpracovává veškeré zadané dílčí úkoly, na úrovni dostačující pro potřeby bakalářské práce. Její význam spočívá hlavně v použití pro studijní účely, případně jako základ pro komplexní zpracování problematiky. Pro případy uplatnění v praxi by bylo nutné uvažovat další prvky a stavět na kompletním a pečlivě stavěném základu.

7 SEZNAM POUŽITÝCH ZDROJŮ

- [1] *Virtuální zprovoznění - budoucnost průmyslu* [online]. 27. 09. 2018 [cit. 2020-10-18]. Dostupné z: <https://www.industryforum.cz/virtualni-zprovozneni-meni-budoucnost-prumyslu>
- [2] Historie PLC. *PLC AUTOMATIZACE* [online]. [cit. 2021-10-10]. Dostupné z: <http://plc-automatizace.cz/knihovna/historie/historie-plc.htm>
- [3] PLC požadavky. *PLC AUTOMATIZACE* [online]. [cit. 2021-5-18]. Dostupné z: <http://plc-automatizace.cz/knihovna/plc/plc-pozadavky.htm>
- [4] Sekvenční (krokové) řízení PLC Simatic ve Step7. *MyIms* [online]. 13.5.2017 [cit. 2021-5-18]. Dostupné z: URL <https://www.mylms.cz/sekvenčni-krokové-řízení-plc-simatic-ve-step7>
- [5] Doba cyklu. *PLC AUTOMATIZACE* [online]. [cit. 2021-5-18]. Dostupné z: <http://plc-automatizace.cz/knihovna/plc/plc-doba-cyklu.htm>
- [6] VOJÁČEK, Antonín. Čím řídit stroje / zařízení - vlastní jednotka, SBC nebo PLC? *Automatizace.hw* [online]. 9. 4. 2012 [cit. 2021-5-18]. Dostupné z: <https://automatizace.hw.cz/cim-ridit-stroje-zarizeni-vlastni-jednotka-sbc-nebo-plc>
- [7] Top Five Vendors in the PLC Market. *Spendedge* [online]. 3.11. 2017 [cit. 2021-5-18]. Dostupné z: <https://www.spendedge.com/blogs/top-five-vendors-plc-market>
- [8] KOZIOREK, Jiří a Libor CHROMČÁK. *Logické systémy řízení a programovatelné automaty* [online]. Ostrava: Vysoká škola báňská - Technická univerzita, [2008] [cit. 2021-5-18]. ISBN 978-80-248-1490-2.
- [9] BOLTON, W. *Programmable Logic Controllers*. 4. Burlington: Elsevier Newnes, 2006. ISBN 9780080462950
- [10] Co by měl každý vědět o programovacích jazycích PLC. *Elektroprůmysl* [online]. 29.8.2017 [cit. 2021-5-18]. Dostupné z: <https://www.elektroprumysl.cz/software/co-by-mel-kazdy-vedet-o-programovacich-jazycich-plc>
- [11] HMI-Automatizace: Displeje. *PLC AUTOMATIZACE* [online]. [cit. 2021-5-18]. Dostupné z: <http://plc-automatizace.cz/knihovna/hmi/displeje.htm>
- [12] HMI-Automatizace: Mechanické ovládací prvky. *PLC AUTOMATIZACE* [online]. [cit. 2021-5-18]. Dostupné z: <http://plc-automatizace.cz/knihovna/hmi/mechanicke.htm>
- [13] 7 Types of Sensors for Object Detection. *Keller Technology Corporation* [online]. 2018 [cit. 2021-5-18]. Dostupné z: <https://www.kellertechnology.com/blog/7-types-of-sensors-for-object-detection/>
- [14] MARTÍNEK, Radislav. *Senzory v průmyslové praxi*. Praha: BEN, 2004. ISBN 80-7300-114-4

- [15] BRAŽINA, Ing. Jakub. *Virtuální zprovoznění výrobního systému* [online]. Brno, 2019 [cit. 2021-5-18]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/116785>
- [16] PARMER, Tim. Be Not Afraid -- the Time has come to Trust Safety rated PLCs. *PCC* [online]. 1.2.2013 [cit. 2021-5-18]. Dostupné z: https://www.pccweb.com/wp-content/uploads/2015/08/Siemens_PLC_Trusting_Safety_PLCs.pdf
- [17] GOBLE, William. Learn to Trust Safety PLCs. *Control design* [online]. 30.5.2003 [cit. 2021-5-18]. Dostupné z: <https://www.controldesign.com/articles/2003/249/>
- [18] KAPILA, Vijay. What's right for you? PLC or Safety PLC? *Bastian Solutions* [online]. 8.1.2014 [cit. 2021-5-18]. Dostupné z: <https://www.bastiansolutions.com/blog/what-s-right-for-you-plc-or-safety-plc/>
- [19] Global industrial robot sales doubled over the past five years. *International federation of robotics* [online]. 18.10. 2018 [cit. 2021-5-18]. Dostupné z: <https://ifr.org/ifr-press-releases/news/global-industrial-robot-sales-doubled-over-the-past-five-years>
- [20] COPE, Kevin. WHAT IS AN HMI? *REALPARS* [online]. 9.4.2018 [cit. 2021-5-18]. Dostupné z: <https://realpars.com/what-is-hmi/>
- [21] Funkční bezpečnost: Předpisy a normy. *TIA na dosah* [online]. 2015 [cit. 2021-5-18]. Dostupné z: <https://www.tianadosah.cz/upload/01-safety-akademie-2019-funkcni-bezpecnost-normy.pdf>
- [22] What is a safety PLC? *PLCdesign* [online]. [cit. 2021-5-18]. Dostupné z: <http://plcdesign.xyz/en/whats-safety-plc/>
- [23] Embedding safety I/Os in ABB robots enhances man-machine collaboration. *ABB* [online]. [cit. 2021-5-18]. Dostupné z: <https://new.abb.com/plc/top-stories/expert/embedding-safety-i-os-in-abb-robots-enhances-man-machine-collaboration>

8 SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK

8.1 Seznam tabulek

<u>Tabulka 1) Seznam alarmů a podmínek jejich aktivace.....</u>	51
--	----

8.2 Seznam obrázků

<u>3-1) Počet světově dodaných robotů do roku 2017.....</u>	12
<u>3-2) Cyklus PLC [5].....</u>	14
<u>3-3) Jazyk příčkového diagramu [10].....</u>	16
<u>3-4) Vizualizace komunikace HMI s PLC.....</u>	18
<u>3-5) Paralelní zapojení dvou výstupních modulů. [22].....</u>	22
<u>4-1) Model zadaného pracoviště v PS.....</u>	24
<u>4-2) Signály poskytnuté s modelem.....</u>	25
<u>4-3) Logický blok pro obráběcí centrum v PS.....</u>	26
<u>4-4) Signály obráběcího centra.....</u>	26
<u>4-5) Signály stacionární brusky.....</u>	27
<u>4-6) Signály upínače a senzory na zakládací a odkládací stanici.....</u>	27
<u>4-7) Funkční a datové bloky a funkce v TiaPortalu.....</u>	28
<u>4-8) Datový blok DB_Memory.....</u>	29
<u>4-9) Aktualizace diod v FB_HMI_Automode.....</u>	30
<u>4-10) Vývojový diagram FB_HMI_Automode.....</u>	31
<u>4-11) Vývojový diagram FB_ProductionRoutine.....</u>	34
<u>4-12) Vývojový diagram FB_RobotProgram – A.....</u>	36
<u>4-13) Přiřazení hodnot pro jednotlivé operace.....</u>	37
<u>4-14) Vývojový diagram FB_RobotProgram – B.....</u>	38
<u>4-15) Vývojový diagram FB_MachineCenter.....</u>	40
<u>4-16) Přiřazení hodnot pro CalledOperation v FB_Conditions.....</u>	41
<u>4-17) Vývojový diagram FB_Conditions-A.....</u>	42
<u>4-18) Vývojový diagram FB_Conditions-B.....</u>	43
<u>4-19) FB_Grinder.....</u>	44
<u>4-20) Ukázka z FC_VariablesAssignment.....</u>	45
<u>4-21) Tag Tables.....</u>	46
<u>4-22) RB001_TagTable.....</u>	46
<u>4-23) Datový blok DB_HMI_Automode.....</u>	47
<u>4-24) Vizualizace HMI panelu.....</u>	48
<u>4-25) Ukázka definování vzhledu informační diody.....</u>	48
<u>4-26) Ukázka aktivace textu z funkce FB_ProductionRoutine.....</u>	49
<u>4-27) Text list pro první informační okno.....</u>	49
<u>4-28) HMI Alarmy panelu.....</u>	50
<u>4-29) Ukázka pop-up screenu pro Stop production.....</u>	52
<u>4-30) Vytvořená PLC instance v PLCSIM Advanced.....</u>	53

<u>4-31) Namapované signály PS.....</u>	53
---	----

8.3 Seznam zkratek

PLC Programmable Logic Controler

PS Proces Simulate

HMI Human-Machine-Interface

SCL Structured command language

9 SEZNAM PŘÍLOH

Nahrávka demonstrující funkci programu

PLC Program